68' MICRO JOURNAL

Australia	A $ 4.75	New Zealand	NZ $ 6.50
Singapore	S $ 9.45	Hong Kong	H $23.50
Malaysia	M $ 9.45	Sweden	30/-SEK

$2.95USA

* Motorola 68020 *
6809-68008-68000-68010

VOLUME VIII ISSUE IX • Devoted to the 68XX User • September 1986

"Small Computers Doing Big Things"

PHOTO CREDIT: NASA

GESSBS-4 $316*

1 MHz 6809 CPU
Sockets for up to 32 Kbytes EPROM
Sockets for up to 16 Kbytes CMOS RAM
One RS 232 serial port
40 TTL Bidirectional I/O lines
4 x 16-bit timers

GESMPU-14 $636*

8 MHz 68010 CPU
Optional 32081 arithmetic unit
Sockets for up to 128 Kbytes EPROM
One RS 232 serial port
4 x 8-bit timers
Real-time clock/calendar and battery

GESMPU-4A $316*

8 MHz 68000 CPU
Sockets for up to 128 Kbytes EPROM
Sockets for up to 64 Kbytes CMOS RAM
One RS 232 serial port
Three 16-bit timers

SOFTWARE

OS-9®, PDOS®, CP/M 68K®, Editor-Assembler, Basic-Pascal-C compilers, FORTH.

Gespac

USA - CANADA
100 West Hoover Ave.
Mesa, AZ 85202
Tel. (602) 962-5559
Telex 386575

INTERNATIONAL
3,chemin des Aulx
CH-1228 Geneva
Tel. (022) 713400
Telex 429989

* 100 piece quantities

Ⓜ is a registered trademark of Motorola Inc.

BOARDS & PARTS FOR GIMIX SYSTEMS

CONTACT GIMIX FOR PRICES, DETAILS, AND REQUIRMENTS FOR HARD DISK AND OTHER MASS STORAGE UPGRADES.

THE GIMIX CLASSY CHASSIS #19 consists of a heavyweight aluminum cabinet, constant voltage ferro-resonant power supply, and SS50 Mother Board with baud rate generator board	$1498.19
#22 Triple Output Regulator Card	$88.22
#93 Baud Rate Generator Board	$88.93
#23 Missing Cycle Detector	$38.23
#92 Filler Plate $14.92 50 Hz Option	$30.00
Cable sets 8" with Back Panel connector	$29.25
for two 8" external drives $44.28 for two 5" drives	$34.98

CPU BOARDS

#01 GMX III CPU & OS-9-GMX III	$1898.01
#02 GMX III CPU & UniFLEX III	$1998.92
The #05 GIMIX 6809 PLUS CPU Board	$570.05
Options: GIMIX OA1 $35.00 9511A	$312.00
SWIP Dal $15.00 9512	$265.00
#03 6800 CPU	$224.03
#06 6800 CPU w/Timers	$288.06
6800 Baud Rate Option	Add $38.00

FLOPPY DISK CONTROLLER

#68 DMA	$388.68

MEMORY BOARDS FOR 6809/68020 SYSTEMS

#72 256xB CMOS STATIC RAM board with battery back up (specify system)	$848.72

MEMORY BOARDS (6800/6809 SYSTEMS ONLY)

#34 8K PROM Card	$98.34
#32 16 Socket PROM/ROM/RAM Board, 24 pin	$238.32
#31 16 Socket Universal Memory Board, 24/28 pin	$268.31

INTELLIGENT I/O PROCESSOR BOARDS

significantly reduce systems overhead by handling routine I/O functions, freeing the host CPU for running user programs. This improves overall system performance and allows user terminals to be run at up to 19.2K baud. For use with GMX III and 020 systems.

#11 3 Port Serial-30 Pin (OS9)	$498.11
#14 3 Port Serial-30 Pin (UniFLEX)	$498.14
#12 Parallel-50 Pin (UniFLEX-020)	$538.12
#13 4 Port Serial-50 Pin (OS9 & UniFLEX-020)	$818.13
#15 24K Version of #11, with either large input or output buffers (specify)	$648.15

I/O BOARDS (6800/6809 SYSTEMS ONLY)

#41 Serial, 1 Port	$88.41
#43 Serial, 2 Port	$128.43
#48 Serial, 8 Port (OS9/FLEX only)	$318.48
#42 Parallel, 2 Port	$88.42
#44 Parallel, 2 Port (Centronics pinout)	$128.44
#45 Parallel, 8 Port (OS9/FLEX only)	$198.45
#58 I/O for RS-232C, 423, 422, w/6850	$244.58
#52 SSDA with 6852	$254.52
#54 ADLC with 6854	$268.54

CABLES FOR I/O BOARDS — SPECIFY BOARD

#95 Cable sets (1 needed per port)	$24.95
#51 Cent. B.P. Cable for #12 & #44	$34.51
#53 Cent. Cable Set	$36.53

OTHER BOARDS & PARTS

#66 Prototyping Board-50 Pin	$56.66
#33 Prototyping Board-30 Pin	$38.33
Windrush EPROM Programmer S30 (OS9/FLEX 6809 only)	$545.00
#76 Video Board-80 x 24	$398.76
#08 Relay Driver Package	$1128.08
#86 Above without Relays	$538.08
Opto Board	$348.05
Binder, 3"	$12.00
Binder, 2"	$9.00

8" DRIVE CABINET & PARTS

2 8" DSDD Drives, Cabinet & Cables 60 Hz only	$1898.08
Cabinet Only for 8" Drive	$848.18
220v/50 Hz Option Add	$30.00
Cable Set-Internal for 2 Drives	$44.82
Cable Set-Internal for 4 Drives	$67.84
Cable from 8" Cab. to Mainframe	$45.01
8" Filler Plate	$14.03

SOFTWARE:

GIMIX exclusive versions of OS-9/GMX I, II, III & FLEX are for GIMIX hardware only. All versions of OS-9 require the #68 controller. When ordered with controller, FLEX is	$30.00
GIMIX versions of FLEX	$90.00
GMX VDisk for FLEX 09	$100.00
GMXBUG: PROMs & Manual	$148.65
Boot or Video/Boot PROMS (6809)	$30.00
GIMIX Boot PROM for UNIFLEX	$50.00
RMS (OS9)	$250.00
DO (OS9)	$70.00
OS-9 GMX III Update w/CPU SPPTROM	$125.00
I/O PROMS w/Update	$40.00
GMXBUG/FLEX/VDISK w/OS-9 III update	Add $175.00
RAM Disk for OS-9	$125.00
O-FLEX	$250.00
OS9 GMX I	$250.00
OS9 GMX II	$500.00
SCULPTOR-6809 (UniFLEX/OS-9)	$995.00
SCULPTOR—68020 (UniFLEX)	$1595.00

CONTACT GIMIX FOR PRICES AND AVAILABILITY OF OPTIONAL UNIFLEX AND OS9 LANGUAGES AND OTHER SOFTWARE.

ALL PRICES ARE F.O.B. CHICAGO.

GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.

LIMITED WARRANTY

GMX INC. ("GMX") Warrants its products against defects in material and workmanship for a period of ninety days from the date of shipment. The obligation of GMX is limited to the repair or replacemnt of any product, free of all charges, which proves defective during this period. This warranty does not cover damage due to accidents, negligence, abuse or tampering.

GMX MAKES NO OTHER WARRANTIES OR GUARANTEES, EXPRESS, STATUTORY, OR IMPLIED, OF ANY KIND WHATSOEVER WITH RESPECT TO ANY PRODUCT PURCHASED, AND ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE IS HEREBY DISCLAIMED BY GMX AND EXCLUDED FROM ANY AGREEMENT MADE BY GMX.

GMX will not be responsibile for any damage of any kind

not covered by the exclusive remedies set forth in this limited warranty. GMX will not be responsible for any special, indirect, or consequential damage caused by its products.

GMX products are not for consumer use. GMX expressly disclaims all warranties on any of its products which may be included in any product normally used for personal or family purposes.

Contact GMX by mail at 1337 West 37th Place, Chicago, IL 60609; or phone at (312) 927-5510; if your product is defective to arrange for its repair or replacement under this warranty.

Repair charges for GIMIX products after warranty period will be $35.00 per hour per board (minimum $35.00) plus parts. Customer pays freight charges both ways. If GIMIX determines that replacement is desirable instead, we will notify you. Charges for checking out complete system will be $500.00 plus parts, freight, and necessary board repairs.

GIMIX, Inc. reserves the right to change pricing, terms, and products specifications at any time without further notice.

GIMIX inc.
1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609
(312) 927-5510 • TWX 910-221-4055

A Member of the CPI Family

68 Micro Journal

6800 6809 68000 68010 68020

10 Years of Dedication to Motorola Users

Editorial Staff

Publisher:
Don Williams Sr.

Executive Editor:
Larry Williams

Production Manager:
Tom Williams

Administration:
Office Manager:
Mary Robertson
Subscriptions:
Joyce Williams

Contributing & Associate Editors:

Ron Anderson
Ron Voigts
Doug Lurie
David Lewis

Dr. E.M. Bud Pass
Peter Dibble
Dr. Theo Elbert
& hundreds more of us

Contents

68 MICRO JOURNAL

"Contribute *Nothing* - Expect *Nothing*" DMW 1986

COMPUTER
PUBLISHING, INC.
"Over a Decade of Service"

"World Wide"

68 MICRO JOURNAL
CPI
Computer Publishing Center
5900 Cassandra Smith Road
PO Box 849
Hixson, TN 37343

Phone (615) 842-4600 - Telex 510 600-6630

68 Micro Journal is published 12 times a year by Computer Publishing, Inc. Second Class Postage paid ISSN 0194-5025 at Hixson, TN and additional entries. Postmaster: send form 3597 to 68 Micro Journal, POB 849, Hixson, TN 37343.

Subscription Rates

1 Year: $24.50 USA, Canada & Mexico $34.00 *a year. Others add* $12.00 *a year surface, airmail add* $48.00 *a year, USA funds!* 2 Years $42.50, 3 Years $64.50 *plus additional postage, for each* additional year.

Items or Articles for Publication

Articles submitted for publication must include authors name, address, telephone number and date, as well as a statement that the material is original and the property of the submitting author. Articles submitted should be on diskette, Macintosh, OS-9 or FLEX format. All printed items should be dark type and satisfactory for photo-reproduction. No blue ink! No hand written articles - please.

Please do not *format with spaces any text indents, chart items, etc. (source listings o.k.) WE will edit in ALL formatting*. Text should be flush left column and use ONLY a carriage return to separate paragraphs or other article text items! MacWrite, FLEX TSC, Stylo formatting acceptable.

Letters & Advertising Copy

Letters to the Editor should be original copy, signed! Letters of gripe as well as praise are acceptable. *We reserve the right to reject any letter to the editor or advertising copy material, for any reason we deem advisable.*

Advertising Rates: Commercial please contact 68 Micro Journal advertising department. Classified ads must be non-commercial. Minimum of $15.50 for first 15 words. Add $.60 per word after the first 15. All classifieds must be pre-paid. No classifieds accepted by telephone.

The VME BUS and OS-9:

Ultimate Software for the Ultimate Bus.

Modularity. Flexibility. High Performance. Future growth. These are probably the prime reasons you chose the VME bus. Why not use the same criteria when selecting your system software? That's why you should take a look at Microware's OS-9/68000 Operating System—it's the perfect match for the VME bus.

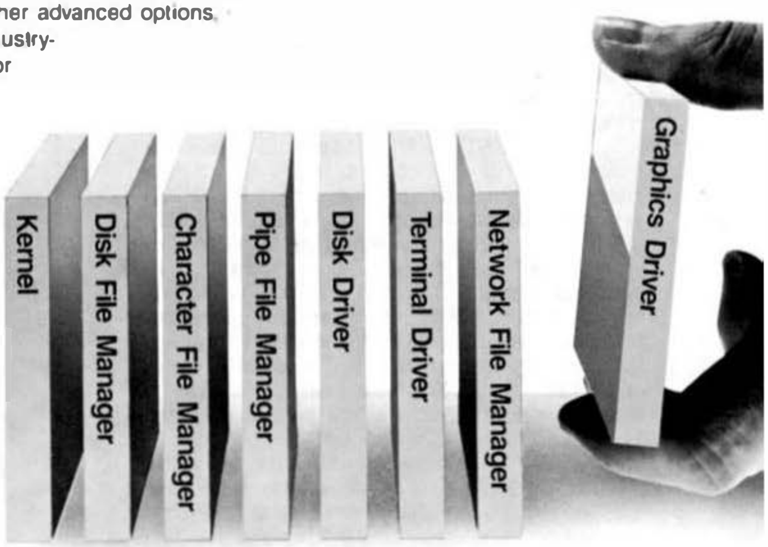
When you're working with VME you must have access to every part of the system. Unlike other operating systems that literally scream KEEP OUT!, OS-9's open architecture invites you to create, adapt, customize and expand. Thanks to its unique modular design, OS-9 naturally fits virtually any system, from simple ROM-based controllers up to large multiuser systems.

And that's just the beginning of the story. OS-9 gives you a complete UNIX-application compatible environment. It is multitasking, real time, and extremely fast. And if you're still not impressed, consider that a complete OS-9 executive and I/O driver package typically fits in less than 24K of RAM or ROM.

Software tools abound for OS-9, including outstanding Microware C, Basic, Fortran, and Pascal compilers. In addition, cross C compilers and cross assemblers are available for VAX systems under Unix or VMS. You can also plug in other advanced options, such as the GSS-DRIVERS™ Virtual Device Interface for industry-standard graphics support, or the OS-9 Network File Manager for high level, hardware-independent networking.

Designed for the most demanding OEM requirements, OS-9's performance and reliability has been proven in an incredible variety of applications. There's nothing like a track record as proof: to date, over 200 OEMs have shipped more than 100,000 OS-9-based systems.

Ask your VME system supplier about OS-9. Or you can install and evaluate OS-9 on your own custom system with a reasonably priced Microware PortPak™. Contact Microware today. We'll send you complete information about OS-9 and a list of quality manufacturers who offer off-the-shelf VME/OS-9 packages.



Modular Hardware Deserves Modular Software

MICROWARE.

Microware Systems Corporation
1866 N.W. 114th Street • Des Moines, Iowa 50322
Phone 515-224-1929 • Telex 910-520-2535

Microware Japan, Ltd.
41-19 Honcho 4-Chome, Funabashi City • Chiba 273,
Japan • Phone 0473 (28) 4493 • Telex 781-299-3122

Micromaster Scandinavian AB
S:t Persgatan 7
Box 1309
S751-43 Uppsala
Sweden
Phone: 018-138595
Telex: 76129

Dr. Rudolf Keil, GmbH
Porphystrasse 15
D-6905 Schriesheim
West Germany
Phone: (0 62 03) 67 41
Telex: 465025

Elsoft AG
Zelweg 12
CH-5405 Baden-Dattwil
Switzerland
Phone: (056) 83-3377
Telex: 828275

Vivaway Ltd.
36-38 John Street
Luton, Bedfordshire, LU1 2JE
United Kingdom
Phone: (0582) 423425
Telex: 825115

Microprocessor Consultants
92 Bynya Road
Palm Beach 2108
NSW Australia
Phone: 02-919-4917

Microdata Soft
97 bis, rue de Colombes
92400 Courbevoie
France
Phone: 1-768-80-80
Telex: 615405

OS-9 is a trademark of Microware and Motorola. PortPak is a trademark of Microware. GSS-Drivers is a trademark of Graphic Software Systems, Inc. VAX and VMS are trademarks of DEC. Unix is a trademark of AT&T.

MUSTANG-020 Super SBC™

DATA-COMP proudly presents the first
Under $5000 "SUPER MICRO".

The MUSTANG-020™

CPI
Mustang-020™
A DATA-COMP
Hi-Speed
Product

MUSTANG-020™

The MUSTANG-020 68020 SBC provides a powerful, compact, 32 bit computer system featuring the "state of the art" Motorola 68020 "super" micro-processor. It comes standard with 2 megabyte of high-speed SIP dynamic RAM, serial and parallel ports, floppy disk controller, a SASI hard disk interface for intelligent hard disk controllers and a battery backed-up time-of-day clock. Provisions are made for the super powerful Motorola MC68881 floating point math co-processor, for heavy math and number crunching applications. An optional network interface uses one serial (four (4) standard, expandable to 20) as a 125/bit per second network channel. Supports as many as 32 nodes.

The MUSTANG-020 is ideally suited to a wide variety of applications. It provides a cost effective alternative to the other MC68020 systems now available. It is an excellent introductory tool to the world of hi-power, hi-speed new generation "super micros". In practical applications it has numerous applications, ranging from scientific to education. It is already being used by government agencies, labs, universities, business and practically every other critical applications center, worldwide, where true multi-user, multi-tasking needs exist. The MUSTANG-020 is UNIX C level V compatible. Where low cost and power is a must, the MUSTANG-020 is the answer, as many have discovered. Proving that price is not the standard for quality!

As a software development station, a general purpose scientific or small to medium business computer, or a super efficient real-time controller in process control, the MUSTANG-020 is the cost effective choice. With the optional MC68881 floating point math co-processor installed, it has the capability of systems costing many times over it's total acquisition cost.

With the DATA-COMP "total package", consisting of a heavy duty metal cabinet, switching power supply with rf/line by-passing, 5 inch DS/DD 80 track floppy, Xebec hard disk controller, 25 megabyte winchester hard disk, four serial RS-232 ports and a UNIX C level V compatible multi-tasking, multi-user operating system, the price is under $5000, w/12.5 megahertz system clock (limited time offer). Most all popular high level languages are available at very reasonable cost. The system is expandable to 20 serial ports, at a cost of less than $65 per port, in multiples of 8 port expansion options.

The system SBC fully populated, quality tested, with 4 serial ports pre-wired and board mounted is available for less that $3000. Quantity discounts are available for OEM and special applications, in quantity. All that is required to bring to complete "system" standards is a cabinet, power supply, disks and operating system. All these are available as separate items from DATA-COMP.

A special version of the Motorola 020-BUG is installed on each board. 020-BUG is a ROM based bebugger package with facilities for downloading and executing user programs from a host system. It includes commands for display and modification of memory, breakpoint capabilities, a powerful assembler/disassemble and numerous system diagnostics. Various 020-BUG system routines, such as I/O handlers are available for user programs.

Normal system speed is 3-4.5 MIPS, with burst up to 10 MIPS, at 16.6 megahertz. Intelligent I/O available for some operating systems.

Hands- n "actual experience sessions", before you buy, are available from DATA-COMP. Call or write for additional information or pricing.

DATA-COMP
Installed Systems World-Wide
OVER 10 YEARS OF DEDICATED QUALITY!

CPI
A Division of
Computer Publishing, Inc.
5900 Cassandra Smith Road
Hixson, Tn 37343
Telephone 615 842-4600
Telex 510 600-6630

```
                    MUSTANG-020 Benchmarks **
                         Time   Seconds

      Type System                          32 bit       Register
                                          Int. Loop     Long Loop

   IBM AT  7300 Xenix Sys 3                 9.7         No Registers
   AT&T 7300 UNIX PC 68010                  7.2         4.3
   DEC VAX 11/780 UNIX Berkley 4.2          3.6         3.2
   DEC VAX 11/750   "     "     "           5.1         3.2
   68008 OS9 68K 8 Mhz                     18.0         9.0
   68000  "   "  10 Mhz                     6.5         4.0
   MUSTANG-020 68020 MC68881 OS9 16 Mhz     2.2         0.88
   MUSTANG-020 68020 MC68881 UNIFLEX "      1.8         1.22

   ** Loop:  Main()
             {
               register long i;
               for (i=0; i < 999999; ++i);
             }

          Estimated MIPS - MUSTANG-020 - 2.5 MIPS
        Motorola Specs: Burst up to 7 - 8 MIPS - 16 Mhz
```

MUSTANG-020™ Software

OS-9

OS-9	$350.00
Basic09	300.00
C Compiler	400.00
Fortran 77	400.00
Microware Pascal	400.00
Omegasoft Pascal	900.00
Stylo-Graph	495.00
Stylo-Spell	195.00
Stylo-Merge	175.00
Stylo-Graph-Spell-Merge	695.00
PAT w/C source	229.00
JUST w/C source	79.95
PAT/JUST Combo	249.50
Sculptor+ (see below)	995.00
COM	125.00

UniFLEX

UniFLEX	$450.00
Screen Editor	150.00
Sort-Merge	200.00
BASIC/PreCompiler	300.00
C Compiler	350.00
COBOL	750.00
CMODEM w/source	100.00
TMODEM w/source	100.00
X-TALK (see Ad)	99.95
Cross Assembler	50.00
Fortran 77	450.00
Sculptor+ (see below)	995.00

Option & Expansions

8 Port expansion RS-232 498.00
(total of 20 serial ports supported)

Expansion for Motorola I/O Channel Modules $195.00

** All Expansion boards:
All expansion boards for old style cabinet will require the 101 expansion cable. Systems ordered with newer PC type cabinet do not require this cable.

101 Expansion Cable $39.95

Sculptor+: We are USA distributors for Sculptor+. Call or write for site or multiple system license & discounts, OEM/Dealer.

Special for complete MUSTANG-020™ system buyers - Sculptor+ $695.00. Save $300.00!

Software Discounts

All MUSTANG-020™ system and board buyers are entitled to discounts on all listed software: 10-70% depending on item. Call or write for quotes. Discounts apply after the sale as well.

MUSTANG-020-

FEATURES

- 12.5 Mhz (optional 16.6 Mhz available) MC68020 full 32-bit wide path processor
- 32-bit wide data and address buses, non-multiplexed
- on chip instruction cache
- object code compatible with all 68XXX family processors
- enhanced instruction set - math co-processor interface
- 68881 math hi-speed floating point co-processor (optional)
  direct extension of full 68020 instruction set
  full support IEEE P754, draft 10.0
  transcendental and other scientific math functions
- 2 Megabyte of SIP RAM (512 x 32 bit organization)
- up to 256K bytes of EPROM (64 x 32 bits)
- 4 Asynchronous serial I/O ports standard
  optional to 20 serial ports
  standard RS-232 interface
  optional network interface
- buffered 8 bit parallel port (1/2 MC68230)
  Centronics type pinout
- expansion connector for add tional I/O devices
  16 bit data path
  256 byte address space
  2 interrupt inputs
  clock and control signals
  Motorola I/O Channel Modules
- time of day clock/calendar w/battery backup
- controller for 2, 5 1/4" floppy disk drives
  single or double side, single or double density
  35 to 80 track selectable (48-96 TPI)
- SASI interface
- programmable periodic interrupt generator
  interrupt rate from micro-seconds to seconds
  highly accurate time base (5 PPM)
- 5 bit sense switch, readable by the CPU
- hardware single-step capability
- mounts directly to a standard 5 1/4" disk drive

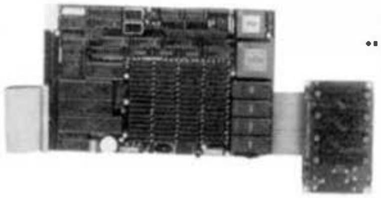
CPI™
Mustang-020-
A DATA-COMP
Hi-Speed
Product

Size 8 15/16 x 5 7/8

These *hi-speed* 68020 systems are presently working at NASA, Atomic Energy Commission, other Government Agencies as well as Universities, Business, Labs, and critical applications centers, Worldwide, where speed, math crunching and multi-user, multi-tasking UNIX C level V compatability and low cost is a *must!*

For a limited time we will offer a $400 trade-in on your old 68XXX SBC. Must be working properly and complete with all software, cables and documentation. Call for more information.

MUSTANG-020 System component prices - Effective July 1, 1986
Prices subject to change - call for latest quotes.



MUSTANG-020 (12.50 Mhz)		$2750.00
** Cabinet	(PC or as shown)	$299.95
5"-80 track floppy	DS/DD	$269.95
Floppy cable		$39.95
OS-9 68K		$350.00
Winchester cable		$39.95
Winchester Drive 25 Mbyte		$895.00
Xebec H/D controller		$395.00
Shipping USA UPS		$20.00
Total:		$5059.80

DISCOUNT LIMITED TIME: Complete System $1061.00

Complete System $3998.80

WE WILL NOT BE UNDERSOLD!

OPTIONS ADD:

UniFLEX	$90.00
MC68881 f/p math processor	$275.00
16.67 Mhz MC68020	$375.00
16.67 Mhz MC68881	$375.00

This price subject to increase
Additional MUSTANG systems soon

Note: Current OS-9 (Ver. 1.2) does not address the MC68881 - Future revisions will. If the 68881 is anticipated in the future, it must be ordered with the system, when originally ordered. UniFLEX does support both the enhanced code of the 68020 and 68881 now.

OPTION BOARDS: ** *Option boards to be installed in Mustang-020 cabinets must be ordered with the extension cable.* The cabinet is too tight for direct plugon. Or specify our new PC type cabinet, with initial order.

PAT - JUST

PAT

With 'C' Source

$229.00

service For Over a Decade! 1,200 Titles .

A Division of

C P I

S.E. Media

5900 Cassandra Smith Rd.
Hixson, Tn 37343
Telephone 615 842-6809
Telex 510 600-6630

6800-6809-68008-68000-68010-68020

PAT FROM S. E. MEDIA -- A FULL FEATURED SCREEN ORIENTED TEXT EDITOR with all the best of PIE. For those who swore by and loved PIE, this is for YOU! All PIE **features & much more!** Too many features to list. And if you don't like ours, change or add your own. **C source included.** Easily configured to your CRT terminal, with special configuration section. No sweat!

68008 - 68000 - 68010 - 68020 OS-9 68K $229.00

COMBO

PAT/JUST

Special $249.00

JUST

JUST from S. E. MEDIA - - Text formatter written by Ron Anderson; for dot matrix printers, provides many unique features. Output formatted to the display. User configurable for adapting to other printers. Comes set-up for Epson MX80 with Graflex. Up to 10 imbedded printer control commands. Compensates for double width printing. Includes normal line width, page numbering, margin, indent, paragraph, space, vertical skip lines, page length, centering, fill, justification, etc. Use with **PAT** or any other text editor. The ONLY stand alone text processor for the 68XXX OS-9 68K, that we have seen. And at a very **LOW PRICE!** Order from: S.E. MEDIA - see catalog this issue.

68008 - 68000 - 68010 - 68020 OS-9 68K
With 'C' source $79.95

SOFTWARE USER NOTES

BY: Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

I warned you all a couple of columns ago that the title of this column might change a little in the near future. I spoke to Don Williams a few days ago and we decided to settle on a very general title so it could follow the current direction of things. The disk file containing these words has the name NOTES80. Since I produced 8 of these before associating myself with 68 Micro Journal, this is the *72nd column that I have written*. That means that I have just ***finished my 6th year of this!!***

The name change in no way implies that I have plans to scrap my old 6809 SWTPc system that runs FLEX. Nor do I have plans for abandoning the discussions of FLEX and software that runs under it. I've been running FLEX for nearly ten years now, and it is an operating system that eminently suits my likes and my style of operation, as well as being suited for the writing and testing of software for stand alone systems as used in instruments and for machine control functions. Unfortunately things in the microcomputer industry do not stand still for very long. The longevity of the 6809 and its cousin the 6502 are certainly a tribute to their design. I know of some builders of instrumentation and related equipment *still using the 6800* in their product. It was and still is quite adequate for a large number of applications.

When the 8 bit processors were designed, THE memory chip was the 2102. It was a 1024 by 1 (i.e. 1024 bit) chip. Eight of them made a 1K memory. The designers didn't forsee the 8K by 8 configurations and couldn't imagine any feasible applications in which 64K of memory would be less than adequate. In my work, we are constantly using microprocessors for more and more sophisticated applications and we are beginning to push 32K of ROM and 8K of RAM. I've heard of this same problem from several other industrial users of the 6809 also. It is not the performance of the 6809 that is becoming the limitation, but the 64K memory limit.

Yes, I know that SWTPc was foresighted enough to include a DAT on their processor board. Software larger than 64K is still a pain because of the necessity to page memory. Virtually no FLEX software takes advantage of the larger memory map that can be utilized by means of the DAT (other than a few "RAM Disk" programs). When I wrote *PAT*, I found that it needed to be about 18K of program, and variable space. It was possible to shoehorn the File Control blocks into parts of the FLEX space, but about 30K was all that could be found for the text buffer. I suppose it would have been possible to write it so that the user could use extended memory and have a whole 64K page for the edit buffer, but the number of users of 6809 systems is already too small to make much of a market for such a product, and to require extended memory would reduce the market to a very few potential users.

At any rate, the low end of the 68xxx products are getting very price competitive with the 8 bit processors. Memory is getting cheaper, and the 68xxx products allow a larger linear (directly addressed without paging) memory map. The 68000 version of PAT is about 28K of code, but there is easily room for an edit buffer of 100K in most systems of 256K memory capacity or more.

All of the above considerations make it imperative at least for my work related projects, to switch to a 68xxx processor and take whatever operating system is available for program development. Lately I have been receiving letters saying things like *"I've been advised to scrap my 6809 systems and buy IBM compatible computers."* This was from someone looking for software for a 6809 system for a magazine subscription label database. How do I answer such questions? Our good 6809 systems started out to be hobbyist systems. At first we were happy to have a working monitor, an Assembler, and the "slowest

BASIC in the West". (Don't knock that old BASIC, it was very good, but not very fast). Most, if not all of our best software came to us from folks who needed something for themselves, and shared it with the rest of us. *They didn't make enough on their software to pay for its development, but having developed it, they received some royalties for their efforts.*

Unfortunately, many more people recognize IBM as a computer manufacturer, than have ever heard of SWTPc, GIMIX, or Smoke Signal. The market is therefore much bigger for software in the IBM and compatible area than it ever was in the 6809 area. The entry of Tandy into the 6809 market with the Color Computer brought hope to a lot of suppliers of 6809 software that the market would expand widely. Unfortunately, Tandy never understood the market and never quite came up with a design that could be expanded upon conveniently, to make something much more useful. Without going into a lot of detail, primarily, Tandy never switched to a true serial port for connection to the outside world (which would have made possible the convenient connection of a "real terminal" to the system.

I've been rambling here for half a page and not indicated where I was going. To sum it all up, like it or not, *I am going to have to move into the 68xxx area with my work related projects.* The company has bought some IBM compatibles simply because of the availability of Wintek's SmArtwork for printed circuit board design, and a number of Computer Aided Design packages for "electronic drafting". We bought a *Mustang 68020 system* and I am satisfied that there will be enough software support for it under OS-9 that we can use it to develop software for stand alone 68xxx systems. Primarily I am pleased with *OmegaSoft Pascal* as a program development tool. It would seem that Certified Software has made it possible to develop software for stand alone systems under OS-9. I will have more to say about this compiler in the future as we gain experience using it. Now I would hope for cross assemblers and compilers to run on the 68020 to develop 6809 s*oftware.*

"Progress" has led me therefore to the point of making this column more of a 68XX(X) user's column. I *will continue to talk about FLEX, but I will add OS-9, STAR-DOS, and whatever else comes along for the 6809 and the 68000 processors.* My emphasis in the past was on the operating system. I slipped into more of an emphasis on the software available to run under FLEX, but your responses to my plea for guidance indicated that many of you wanted more basic information on FLEX again, so I have complied with that request recently.

I've said all along that I have little time for any deep research for the material in this column, so that what is written here must somehow relate to what I am doing presently. A good deal of this is "off the top of my head" and it is intended to be just that, and not to be taken too seriously... (Over the years, a few readers have taken my remarks *MUCH* too seriously. One reader even chided me for saying bad things about FORTH because "You have more influence than you realize with your readers".) I give the readers of this column a lot of opinion. I have always assumed that they were smart enough to think for themselves and sort out what was fact and what was opinion by trying things out for themselves. *I do not consider myself as having or being the last word on anything.*

I must add one further note to this and then I will get on with the more usual topics of this column. I have missed a monthly deadline for the first time in my 6 years. It was because of business and work related problems, and it is not likely to happen again. Along with the missing of the deadline, as you might imagine, I missed answering a great deal of correspondence. I am now in a position to catch up a little, but I don't expect ever to get through all of my back correspondence. If you wrote me a letter in the past few months and are awaiting a response, you might try again. I will answer all correspondence having to do with configurng PAT, or any problems reported to me concerning its use or configuration for various terminals or computer hardware. In fact, most or all such questions should have been answered before this appears in print. I am sorry for the lapse on my part. It was totally unavoidable.

Pat Catastrophes

I have been trying to get the last bugs out of PAT so that the "release disk" would present you a functional PAT without a lot of strain and effort. So far my efforts have been totally foiled by Murphy's laws. As soon as PAT hit the market, I received five or six letters from purchasers with configuration problems. First, the oldest version of FLEX (FLEX 9.0) lacks the INCHNE vector at $D3E5. I wrote a special PATF9 version that talks directly to the terminal port. Then someone with an ANSI terminal wrote. I had to do a special version of PAT called PATANSI for those applications, with longer terminal control strings and a special routine for cursor addressing. In spite of a lot of pre-testing by several people including myself, a few more bugs have been detected. There are presently no known uncured bugs, but a few more may still be hiding somewhere. At any rate, *the last release disk that I prepared for Southeast Media was version 2.5*. It turns out that there was an error in just ONE of the terminal configure files called TERM.TTX, which I had copied to TERM.LIB, and used to compile all the versions of PAT on that release disk. An attempt to append any of the other terminal configuration files causes great problems.

Unfortunately I supplied only the text files for the terminal configuration files, which must be assembled to the .BIN files before appending them to the PAT.BIN file. Assembling the file is easy for most of the users, though the "startup" chapter of the manual doesn't indicate that the .TXT file has to be assembled first, this is discussed in appendix A. I had intended to supply the .BIN files too. If you have version 2.5 and have the TSC assembler, you can fix the terminal configuration files by adding one byte to the TINIT string. Just add one more ",0" to the end of the FCB list at the label TINIT, and assemble this file and append it to PAT, PATANSI, or PATF9 to make your command file.

If you have PL/9 go at it the other way. That is, edit the appropriate TERM.XXX file to match your terminal, copy it to TERM.LIB, and recompile whatever version of PAT you need.

I promise that the next version will have more bugs removed, and that I won't issue it until I have checked and tested it 23 ways. I have an idea for a better way to do the terminal configuration, and I am working on it. The result will be that there won't be a need for several versions of PAT.BIN.

There are a couple of bugs in version 2.5 that have been cured for the next update (Version 2.7). If you type ESC ^G without typing a line number to go to, PAT will be confused and try to go to line 0, which doesn't exist. The cursor will stick on the status line and the indicated line number will be 0. A control T will get you out of the funny state. If you do a backward search for something that is on line 1, you might also run into this problem. The cure is the same. Finally, if you load a file that happens to have line 23 longer than 80 characters, PAT will get out of sync. Again Control T will cure things. These will all work properly in the next update.

I have one improvement for the next update of PAT that I think will be liked by those who use it to edit programs. When you program in "C", Pascal or PL/9 you sometimes change the structure of a program and need to shift a block of program lines to change the indentation level. I have added ESC Sn <CR> to PAT. It is a BLOCK function, and is added to the block copy move and delete functions. First you mark a block of lines, and then use the above function to shift all the marked lines n spaces to the right (by inserting spaces at the start of each line). If n is negative, (eg. -5), spaces will be deleted from the start of each line, effectively shifting them to the left. The function is smart, and it will not delete printable characters from the start of a line. It stops deleting when it runs out of spaces. However, the user must not shift a line so far to the right that it overflows, since PAT doesn't check for that condition. Perhaps I will make the function a little smarter before the next update.

PAT, OS-9 and CT-8212

Sounds like a good combination, or so I thought. I have PAT running (with a few rough edges) under OS-9/68K on the Mustang, and had sent Don Williams a copy configured for the CT-8212. Don reported a few problems so I dusted off the old CT-8212 and connected it to the Mustang. I found a couple of dumb errors in my configuration file immediately, and then discovered a serious problem of compatibility. For a terminal to run a screen editor, it is necessary for the computer to be able to send the terminal instructions to position the cursor at any line or column of the screen. When I tried using the CT with PAT, I had a couple of singularly puzzling problems. I would cursor down the screen and suddenly the cursor would jump to the incorrect line. I would go down one more line and all would be OK.

I had seen such problems before, and I knew roughly where to look. Most terminals use an "escape sequence" to put the cursor at a particular place on the screen. For example the ADM terminals, the ADS, and at least some of the Televideo terminals use ESC '= Y X, a four character sequence in which Y and X are the ROW and Column in that order, but they are transmitted as a character whose ASCII value is that of Y or X with $20 added. That is ROW 0 (the top line) is $20. The next line is $21, etc. Same for columns. The reason for the addition of $20, is that all ASCII codes less than that value are control codes, such as CR and LF, and in fact *therein lies the problem with the CT-8212 and OS-9/68K. OS-9 has provision for adding a LF after each CR in the output stream.* That is useful for terminals, where a CR from the terminal is echoed CRLF, and when reading a text file in which ends of lines are marked by CR only. The OS adds the LF. Well, the CT-8212 doesn't use the $20 offset, so the code for line 13 is $0D, a CR. OS-9 obediently adds LF which replaces the column information, so the cursor goes to the wrong column, etc.

OS-9, it turns out, has a utility called tmode that allows setting of parameters for terminals, much like TTYSET of FLEX. It has one parameter that may be represented by "lf" or "nolf". The value "nolf" is supposed to suppress the LF that is automatically tacked on after a CR. Well, it is not difficult to set the mode to "nolf" before a cursor positioning instruction and to restore "lf" mode afterward, but that didn't cure the entire problem. *It seems that OS-9 also tacks a CR onto an LF. That is, when an LF is sent, a CR is added.* That feature doesn't switch off when "nolf" mode is invoked! With nolf mode, I could move the cursor down into and past line 13, but I couldn't get it on line 10!! I think this is a bug in OS-9. If you wanted to suppress the autoLF after a CR, you would probably want to eliminate the autoCR after LF. *Since Microware wants $250 a year for the privilege of calling them and asking a "technical question", all I can do at this point is to submit a bug report and wait to see what happens.*

At this point in my familiarity with OS-9 (or rather unfamiliarity with it) I would not attempt to write a device descriptor let alone get inside OS-9 and find the device driver that correctly looks at the nolf flag for one condition and not the other. I'll just have to wait for an answer.

CT-8212 More

I wrote the above a couple of weeks ago. Since then I decided that I vaguely remembered that the CT-8212 has a command to set the cursor positioning offset value to something other than zero. I checked the manual and sure enough, there it was. Just send the terminal ^_ and ^J followed by the offset character, a space in my case since I thought it would be nice to get out of sending control characters to the terminal. That ought to work fine except that control J happens to be the same as a linefeed, and it is the code that causes OS-9 to add the CRLF. You would think that would mean that one could set the offset to 13, and that perhaps that would work. For some reason, it didn't. I finally figured out that if I turned on the CT-8212, set it to Half Duplex mode, disconnected it from the computer and typed ^_ ^J space, I could set the mode I want. I made a terminal configuration file that uses the $20 offset and configured the terminal manually. It works, but I would not call it an acceptable way to go. It is too much bother to disconnect the terminal to set its mode. *Microware, is there a bug in OS-9 68K, or is it* *supposed to work that way?*

An Operating System can be helpful and useful if it does functions (as OS-9 does) to monitor the input channel for ^C as an "interrupt" of the program, adds LF to CR, has a couple keys that function as XON and XOFF for the terminal to stop and start data display, etc. but in order to do certain jobs (such as work with a terminal like the CT-8212 with zero offset for cursor positioning) all of the little OS features have to be switchable so that they can be disabled and the OS can become completely passive, giving the programmer complete control over what is sent to the terminal or printer. Until this present discovery, I was convinced that OS-9 was capable of being unobtrusive. Without a fix for this problem, it is not usable with any terminal that uses no offset for cursor positioning codes.

The Ultimate Seminar

I received a notice of a Two Day Seminar the other day. I've expressed my regrets here that the bookstores are full of books in the vein of "Getting the Most Out of Wordstar" rather than "Programming the 68000". The seminar title is "Using Lotus 1-2-3". The cost of the two-day program is $400. Since Lotus 1-2-3 at last check sells for $250 or so, it would seem to me to be the ultimate insult to a purchaser to imply that he had to spend almost twice the cost of the software to learn how to use it! It has always been my observation that I couldn't possibly learn enough to be useful in two days. A $20 book on the subject at hand is generally moie beneficial than a $300 Seminar or crash course. We in the U.S. seem to fall for anything that is the quick and easy way out. "Loose up to 30 pounds the first month, and eat all you want....." Become an expert at running your computer in two days.... Sorry, but it just doesn't work that way. *In my experience, "instant" knowledge is forgotten the next instant.*

Bargain

Tandy is good for something after all. They had one of their Warehouse Sale days here recently and thee advertized DT-1 terminals for $99 each. Someone from our company was there when they opened the doors, and managed to get a couple. They have the sharpest monitor I have ever used. All the characters are crisp from corner to corner on the screen. They have reverse video, and half intensity that is really about 0.97 intensity. that is, you can hardly see a difference. They run fine at 9600 baud and are supposed to run at 19.2K baud, but so far we haven't been successfull in our efforts to run them at that rate. I have configured PAT to run on them, and that brings me to a short discussion.

I configured PAT to display the status line in half intensity mode on teiminals that support that mode without using a character position on the screen to turn it on and off. The DT-1 has reverse video that does use that position for attribute on and another for attribute off. I decided to get the problems worked out so PAT would work with either type of terminal I found that those that use the character position are what I call MODAL in operation. You set up an area of the screen in reverse video, and all you have to do is position the cursor within that area and anything you output is in reverse video. That is, the attribute is tied to the screen position and not the characters that follow the attribute-on code. After getting PAT working with both types of attribute change schemes, I must say that I like the MODAL arrangement better. I can bump the cursor down to the status line and update the line number without sending those set and clear attribute codes. It is much faster, and the screen updates faster when wordwrap occurs.

The next update version of PAT will have this change included. I have also added a control code ESC ^B to toggle the "BELL" on and off. I was working late one night and the BEEP got annoying. Some terminals are so loud, (and I had one of those) that I thought I would wake up everyone in the house at the end of every line, so I added the toggle. PAT comes up with the bell ON, but you can turn it off. There is no indicator on the status line, but you know at the end of every line which mode is operative.

+++

C User Notes™

By: E. M. (Bud) Pass Ph.D
Computer Systems Consultants
1454 Latta Lane, N.W.
Conyers, Ga 30207
404 483-1717/4570

INTRODUCTION

This chapter begins the discussion of the proposed ANSI C standard and the discussion of common problem areas in the use of the C language.

TSC, Windrush, and Microware have all sent information to update me on their latest changes and enhancements to their products. If others have information of interest to the readers, please send it. This is one of the best sources of input.

PROPOSED ANSI C STANDARD

The ANSI (American National Standards Institute) X3-J11 committee has published several versions of a draft of a new standard for the C language. If approved, it would not be binding on any C compiler writers, but compliance to it could definitely become a selling point for a given C compiler. Compilers based upon the complete standard cannot possibly appear before the planned completion date of December 1986 and probably will appear only substantially later, although compilers being incorrectly advertised as conforming to the full ANSI C set of standards have already begun to appear. In addition, standards typically evolve in response to changing environments, so there will probably be multiple standards for some period of time (ANSI C86, ANSI C87, ...?).

When the K & R white book was written (1978), almost all C compilers were UNIX-based or directly derived from a UNIX-based system. The K & R book has never been updated, despite the number of ambiguities in it and despite the changes which have occurred in the mini and micro computer industries since that time. Small C implementations which grew up to be full C implementations almost never are completely compatible with the UNIX V7 C "standard", which itself has evolved since 1978, into UNIX V3, UNIX V4.2BSD, and the current UNIX V5 C compilers. Thus, the need for a new standard for the C language was recognized several years ago.

The discussion below begins the presentation of the major points covered by the proposed standard and of the differences between the current "standard" C compilers (more or less based on Kernighan and Ritchie) and the new standard.

All C programmers will eventually be affected by this new standard. The degree to which a programmer will be affected will depend upon several factors. If they are porting programs from older implementations, they may be seriously and adversely affected. If they are writing programs on a conforming compiler with the purpose of porting to other conforming compilers, they should be beneficially affected with having an easier job.

The existence of the new standard does not imply complete portability among conforming implementations. It does imply that the implementations will act in certain prescribed manners in specific areas, but it allows many implementation dependent areas, including local extensions which are not portable. Some of the more important implementation dependencies are the length of variables (are ints 16, 18, 24, 32, 36, 48, or 64 bits long?), the naming conventions for disk files, the system requirements on external names (8 characters, upper case?), the maximum sizes of program and data spaces, peculiarities of library functions, and format of operating system command lines.

The overall effect of the new standard on existing C programs is unknown. However, treating placement of older programs on conforming compilers as porting problems, the better-written programs will generally continue to be easier to port than the poorer-written ones. There will probably be a series of programs introduced commercially to translate programs written for commonly-used C compilers to the new standard.

The major changes and enhancements in the new standard are as follows:

tighter data typing
tighter pointer usage checking
operator order precedence changes
new data types
function prototyping
trigraphs
new constant formats
identifier lengths
aggregate initialization
new preprocessor formats
library standardization
variable number of function arguments

These changes will be discussed in later chapters.

C PROBLEM AREAS

The C language is powerful and rich enough to cause problems to even the most experienced programmers. The items below begin the presentation of some of the types of problems commonly encountered in the use of the C language.

Sloppy Formatting

Since the C language is generally free-format (as opposed to BASIC, FORTRAN, and other popular languages), a C programmer is free to place multiple statements on one line, indent statements in a random fashion, and generally ignore good formatting concepts.

This can lead to difficult debugging and maintenance of such programs. Luckily, programs to format C programs are readily available and should be used as required. Of course, they would not have been written had not they been needed.

Overuse of Goto's

The C language supports several types of structured programming concepts, such as if, while, do, and blocks. It also supports goto statements, about which there is much controversy. In 1968 Dijkstra, a Dutch computer scientist, published an article in which he condemned the use of the goto statement. He noted that goto's, improperly used, can destroy the structure of an otherwise well-structured program, and make it impossible to prove that a program performs correctly.

Since that time, there have been many articles written from from both pro- and anti- goto statement standpoints. The overuse of goto's is condemned almost universally. However, some people would like to completely prohibit them. This discussion has all the elements of a religious debate, with fanatical adherents to both sides.

The use of goto statements is never necessary; however, in some cases, selective use of them may produce clearer programs than their avoidance. For example, the C language provides the break statement to exit from the current level of for, switch, and similar structures; however, it provides no automatic means to immediately exit from a nested for structure. The goto statement provides this facility in a simple manner without the necessity of adding flags to indicate the exit.

Unfortunately, goto statements can be used to violate program structure, such as branching into a loop or branching from one inner loop to an unrelated inner loop. They may also turn an otherwise simple program into a "bowl of spaghetti" if they are overused. Most programmers who have ever maintained an existing system of programs have encountered this class of programs, usually with dread, because of the problems associated with the modification of such programs. Even if they appear to function correctly, often even small changes are difficult to make properly. Sometimes it is easier to rewrite such programs than to try to maintain them.

Uninitialized Variables

Although global and static local variables are initialized to zero, automatic local variables are not necessarily initialized to any particular value. In fact, since local variables are normally allocated on a stack, they will usually contain random values. Failure to initialize variables before use may cause intermittent failures and maintenance difficulties.

This situation is complicated somewhat by restrictions placed on initialization of automatic variables by K & R and compiler implementations. For example, the following initialization is (at least theoretically) illegal inside a function or a block:

```
   char hex[] = {"0123456789abcdef"};
```

Misuse of Pointers

Although pointers are themselves variables, the potential damage done by them is often much greater than that done by ordinary variables. In order to use a pointer successfully, it must be declared properly, initialized correctly, and manipulated properly, and the object being pointed to by the pointer must also be declared properly, initialized correctly, and manipulated properly.

On systems without memory protection, pointers improperly initialized or used may modify program data or code or parts of the operating system, possibly leading to problems difficult to locate and correct or totally hidden until the program is ported to another compiler or system.

Even on systems with memory protection, pointers may not access or modify the expected data, also leading to hidden problems, although program code and other programs' data should be secure. Many systems with memory protection also require alignment to even or quad byte addresses, further complicating the problems of hidden bugs.

Array Declaration And Use

Although the size of an array is declared assuming base one, array subscripts are used assuming base zero. This inconsistency sometimes catches even experienced C programmers. For example,

```
   #define LEN 10
   int i, x[LEN];
```

```
 :
 for (i = 0; i <= LEN; ++i)
    x[i] = 0;
```

clears eleven, not ten, elements in array x.

C PROBLEM

Following are two unrelated short C functions. Neither function always works correctly, although both compile and execute. Explain what is wrong with them and fix them.

This function writes garbage characters to a file, rather than the values of the characters passed to it.

```
int outchar(fd, ch)
int fd;
char ch;
{
  return write(fd, ch, 1);
}
```

This function works on some compilers and not on others.

```
int itexists(filename) /* return 1 if file exists */
char *filename;       /* otherwise return 0     */
{
  FILE *fp;
  int i;

   if (i = ((fp = fopen(filename, "r")) != NULL))
     fclose(fp);
  return i;
}
```

A "quickie": what does "x+++++x" mean, if anything?

EXAMPLE C PROGRAM

Following is this month's example C program; it computes the interest on a loan and prints an amortization schedule.

```
/*
 * Print amortization schedule for fixed-payment loan.
 *
 * amount   = principal borrowed ($)
 * annual   = annual percentage rate of interest (%)
 * term     = length of loan (years)
 * month    = month (1-12) loan is made; default = now
 * year     = year (00-99) loan is made; default = now
 * interval = months between payments (1,2,3,4,6,12);
 *          default = 1 (monthly)
 *
 * Programmed by Dean Douthat
 */
#include <stdio.h>
#include <time.h> /* for UNIX and compatibles */

#define PAUSE 0   /* 1: single sheets, 0: not */

static int month, year;
static float amount, annual;
static int term, interval, periods, pmts_per_year;
static long int balance, payment, interest;
static double periodic;

main(argc, argv)
int argc;
char *argv[];
{
  double x, r;
  char *malloc();

  if(argc < 4)
  {
    fprintf(stderr, "%s %s %s\n",
      "Usage:", argv[0],
      "amount annual term [month year [interval]]",
    exit(1);
  }
  sscanf(argv[1], "%f", &amount);
   if(amount < 0.0) error("Illegal amount");
  sscanf(argv[2], "%f", &annual);
  if(annual < 0.0 || annual > 100.0)
    error("Illegal annual percentage");
  sscanf(argv[3], "%d", &term);
   if(term <= 0) error("Illegal term");
  if(argc >= 6)
  {
    sscanf(argv[4], "%d", &month);
    if(month < 1 || month > 12)
```

SOMETHING FOR ALL OF US

FROM ALL OF US

```
    error("Illegal month");
sscanf(argv[5], "%d", &year);
if(year < 0 || year > 99)
    error("Illegal year");
if(argc >= 7)
{
    sscanf(argv[6], "%d", &interval);
    if(interval < 1 || interval > 12 ||
            (12 % interval))
             error("Illegal interval");
      }
      else
         interval = 1;
    }
    else
    {
       today();
       interval = 1;
    }
    pmts_per_year = 12/interval;
    periodic = annual/(pmts_per_year * 100.0);
    periods = term * pmts_per_year;
    balance = amount * 100.0;
/*

    Find periodic payment amount
*/
    r = 1.0 + periodic;
    for(x = 1.0, i = 0; i < periods; ++i) x *= r;
    payment = (long)
       (100.0 * amount * periodic * x/(x - 1.0) + 0.5);
/*
    Compute amortization table values
*/
    for(i = 1; i <= periods; ++i)
    {
       interest = (long) (balance * periodic + 0.5);
       if(last = (i == periods))
          payment = interest + balance;
       line = lineout(i, line, last, PAUSE);
    }
 }
 static long cum, ytd;
 static int omonth, oyear;
 static char *month_name[] =
```

```
{
   "BAD",
   "Jan",
   "Feb",
   "Mar",
   "Apr",
   "May",
   "Jun",
   "Jul",
   "Aug",
   "Sep",
   "Oct",
   "Nov",
   "Dec"
};

lineout(i, line, last, pause)
int i, line, last, pause;
{
   long princ_pay;

   if(line < 4)
   {
      omonth = month;
     oyear = year;
     cum = 0.0;
      line = header(pause);
   }
    if((month += interval) > 12)
   {
        ytd = 0.0;
        ++year;
        month -= 12;
         if(interval < 12)
        {
            putchar('\n');
           ++line;
        }
         if(line + pmts_per_year > 61)
            line = header(pause);
    }
    princ_pay = payment - interest;
    balance -= princ_pay;
    cum += interest;
    ytd += interest;
   printf("%5d %s %02d",
      i, month_name[month], year % 100);
   printf("%12.2f%12.2f%15.2f%15.2f%13.2f\n",
      princ_pay/100.0, interest/100.0,
      balance/100.0, cum/100.0, ytd/100.0);
   ++line;
   if(last)
   {
      printf("\n   Last Payment = $%.2f",
        payment/100.0);
      if(!pause)
         printf("\f\f");
   }
   return(line);
}
static char *type[] =
{
  "BAD",
  "monthly",
  "bimonthly",
  "quarterly",
  "thrice-yearly",
  "BAD",
  "semi-annual",
  "BAD",
  "BAD",
  "BAD",
  "BAD",
  "BAD",
  "annual"
};

header(pause)
int pause;
{
   if(pause)
   {
      fprintf(stderr,
         "Insert paper then hit RETURN ->");
      getchar();
   }
   else
      putchar('\f');
   printf("\n\n  $%.2f at %2.3fannually \
for %d years starting %s %d\n",
      amount,annual,term,month_name[omonth],oyear);
   printf("  in %d %s payments of $%.2f",
      periods,type[interval],
        payment/100.0);
   printf("\n\n              ");
   printf("Principal   Interest       Balance\
      Cum Int     YTD Int");
   putchar('\n');
   return(7);
}
today()
{
   struct tm *localtime(), *now;
   long clock, time();

   clock = time(NULL);
   now = localtime(&clock);
   month = now->tm_mon + 1;
   year = now->tm_year;
   if(month <= 0 || month > 12)
      month = 0;
   year%= 100;
}
error(s)
char *s;
{
   fprintf(stderr, "%s\n",s);
   exit(2);
}
```

+++

Basically OS-9™

By: Ron Voigts
2024 Baldwin Court
Glendale, Il 60139

NEW OS-9, OLD OS-9

I sometimes feel like I'm living in a time warp. This is the September column, but I am writing it at the end of April. I'll send it in at the end of May. Probably by July they'll be putting together the magazine, so it can be mailed out in August. What I write now, seems like a long time until it get published. A current issue may be an old story by the time it gets to you. But I feel this months topic is relatively important that a few months lag will not detract from its importance.

Earlier this month I received my upgrade to OS-9 for the Color Computer, Version 2.00.00. It has occurred to me that by September many readers will already have theirs. I think many will not and there will many more who are interested in what has been changed or improved. This version is important enough that it was given an entirely new revision number.

Some of the older commands have been modified or revised. A few new ones have been added. There have been changes to the system. GETSTAT and SETSTAT have been updated. The system call, VIRQ, has been added. A whole bunch of devices can now be added to the system.

The changes to the system include some minor and some major ones. The minor ones include the following. An auto repeat key feature has been added. Hold down a key and after about a second it will start repeating. Now a beep occurs if a control-G is sent from the keyboard. From Basic09, a CHR$(7) can be used. The <@> key is now the ALT key. Holding it down with another key will set the 8th bit. The letter A is $41. With the ALT key, it becomes $C1, creating a low resolution graphics character. The error number $DC is now HANG UP, replacing the ILLEGAL BLOCK SIZE. The descriptors for drives 2 and 3 are not part of the initial Boot file, but can be added later.

A major change was with terminal I/O. It is now divided into three parts. There is is still TERM, but video output is handled through subroutine modules. Either there is CO32 for the normal Color Computer screen or CO80 for an 80 column screen. I found, CO80 does not work with my WordPakII and I strongly believe it will only work with a Radio Shack 80 column card, *should one ever come along*. There is also graphics module, GRFO. It is only needed if high resolution graphics are intended.

Graphics have had some new additions. Some of the notable additions are DRAW and ERASE CIRCLE, and FLOOD FILL which works somewhat like the old PAINT command. Also additional graphics buffers may be allocated and selected. These little additions should make writing high resolution graphics easy from almost any language.

The GETSTAT and SETSTAT have had some additions. For GETSTAT there is the following. SS.DevNm will return the device name. SS.KySns returns key down information. This includes the the SHIFT, arrow keys, @, Cntrl-Clear and the spacebar. And SS.ComSt returns information for SCF devices like parity, stop bits, word length and baud rate. I$SETSTT now has the following. SS.KySns enables and disables the keyboard sense ability. SS.ComSt will set information for SCF devices. SS.AAGBf allocates additional graphics buffers and SS.SLGBf selects the graphics buffer.

There has also been additional device descriptors and drivers added. To name a few of them, SSC and SSCPAK support the Speech/Sound Cartridge. M2 and MODPAK are for a virtual driven modem. And H0 (or H1) and CCHDisk can be used for a hard disk. Running a hard disk sounds like a great idea, but I seriously doubt that anyone will spend $200 for a computer and then put $1500 into a hard disk.

New commands have been added and old ones modified. There four new ones. CONFIG is one of the new ones that provides a menu driven method to customize systems disks. INIZ forces the allocation of device buffers. HELP is an on line help feature. TUNEPORT lets you find the optimum delay loop for SCF devices. FORMAT now lets you format disks without prompts. Actually it did before, but it was never documented. OS9GEN has a single drive option. TMODE and XMODE can adjust for 32 or 80 character columns. Some of the other commands have been modified to use either 32 or 80 column displays. They include CONFIG, DIR, DUMP, LOGIN, MDIR, PROCS, TMODE and XMODE.

There are many more thing I have not covered. Radio Shack supplies an 80 page manual describing them. I am glad to see some of the changes. It is nice that the graphics has been given a better "deal". Also, I like some of the frills like the auto repeat key and the Cntrl-G. But, if you do upgrade, you may find that third party software does not work. You'll have to go back and upgrade with them. Most vendors will do so for a small fee.

TO FRACTURE MEMORY WITHOUT REALLY TRYING

This part concerns Level I users, but the Level II users will find it interesting and imformative, too. If you run Level I, you must be extra when using memory. Usually when a program is loaded it is put into the highest memory locations and the data area for it is allocated from the lower areas. The result is that the center area of memory is left open. Should a second program be loaded, it would be given memory by the same treatment. Now remove the first program, using UNLINK or perhaps it is simply finished. The second program still occupies its original memory location and we find memory is in a few chunks, instead of one continuous area. This could make it difficult to load something large. Trying to do so would result in a Error #207. The best advice would be to UNLINK the second program and then reload it if it was needed.

What do you do if the system fractures memory on its own? All devices need a buffer area. This space is allocated to them when they are placed into the Device Table. And devices are first placed into the table when they are used. Another side note, the buffer area is allocated from a high area in RAM. Let's say you load BASIC09. It occupies memory starting at $2A00. Next you dump a listing to the printer. This is the first time you've used the printer since booting up. So the printer gets its buffer area starting at $2900. Later you say, "BYE" to BASIC09 and return to OS9. *Now, your memory is fractured with a page gone at $2900.* Fortunately, there are small fragments of free memory at the high end of memory that would probably be taken first. But this is not always the case.

It might be helpful to examine the device table and understand how it works. Going back to the direct page pointers, is the Device Table Pointer, otherwise called D.DevTbl. This points to a table containing 9 byte entries of each device being used in the system. The format of each entry is as follow:

Name	Bytes	Purpose
V$DRIV	2	Device Driver
V$STAT	2	Static Storage
V$DESC	2	Device Descriptor
V$FMGR	2	File Manager
V$USRS	1	User Count

To show how each of these is used let me let you take a look at my table. It is located at $B56D. This the value stored at D.DevTbl. I used DEBUG to examine the table. Following is what was there with a little rearranging.

```
B56D - B600 B300 BACE CFFE FE
B576 - BC35 B100 C47C CBA0 02
B57F - EE13 A200 EDE6 CFB6 04
B588 - B600 B300 BAFD CFB6 00
B591 - 0000 0000 0000 0000 00
```

The first number is the location in memory. The next 9 bytes are a device's entry. The last one with all 0's is the end of the table. It is best to examine one of them. Look at the last entry, starting at $B588. $B600 is the start of my disk driver SDISK. $B300 is the buffer space allotted to it. $BAFD is the start of the device's location. $CFB6 is the file manager, SCF. And the last $00 is the user count. If I had something running on it while I was examining the table the count would have been $01.

The trick to avoiding memory fracture is to get your device in the table early. Otherwise, it may result with the static storage being allocated from somewhere in middle memory, which is not to desirable. A good way to get something in the table is to use it early. For example, having in you 'startup' file:

```
display 7 >/p
```

will put the printer device 'p' into the device driver table. The new version 2.00.00 of OS-9 has a utility called, INIZ. It adds, via the OS-9 system call, I$Attach, the device to the table. For example, to use it to on the printer and disk drive 1, you would enter:

```
iniz p d1
```

A call is made to I$Attach and they are entered into the device table. The user count is also increased for each related module. So, when attach the printer, P, PRINTER, and SCF have their user count increased by one.

I find one flaw in the utility. It works fine, but it's format does not conform to OS-9 convention. Ideally when addressing a device, a slash is prefixed onto its name. Psychologically it becomes second nature. INIZ does not use this, but rather prefers to have its devices referred simply by name. When typing, I enter names like /P, /D1, /D2 and so on. This month I offer an variation of the program to put devices into the systems.

It is called ATTACH. It still uses the I$Attach system call. The difference is I$Parse is used to examine the command line. It goes thru the command line and finds legal OS-9 names. If it encounters a slash, it moves to the next position. It recognizes spaces and commas as separators for names. Using ATTACH is very simple. All of the following do the same thing.

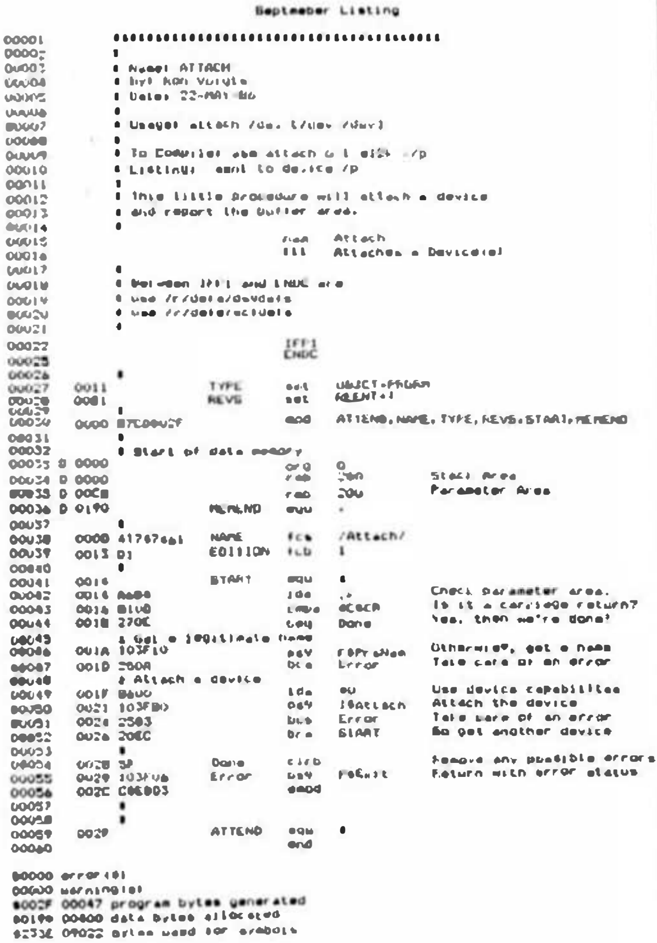
```
attach p d1
attach /p /d1
attach p,d1
attach /p,/d1
```

No matter which way it is entered, drive one and the printer will be put into the device table and buffer space will be given to it.

OS-9 NEW COMERS

This month marks two computers joining the ranks of OS-9. First, there is a plug-in board for the IBM-PC. The board's CPU is a 68010 with a real time clock, extra RAM, serial ports and parallel ports. With it the IBM-PC will be able to run OS-9. The other new comer is OS-9 for the Atari-ST. The Atari line has two 68000 machines, the 520 and 1040. *Both systems are from TLM Systems, Fresno CA.*

Besides systems being adapted to run OS-9, there are ones coming made to run it. Earlier this year, DATA-COMP introduced the MUSTANG-020™. It runs a 68020 CPU with a 12.5 (16.67 available on special order) MHz clock. The MUSTANG-020 runs OS-9 and supports all the available software. So, whether it is on an old machine or a new machine, look out world here comes OS-9!



```
                        September Listing

                *****************************************
                *
                * Name: ATTACH
                * by: Ron Voigts
                * Date: 22-MAY-86
                *
                * Usage: attach /dev [/dev /dev]
                *
                * To Compile: asm attach o l e=$24 >/p
                * Listing: send to device /p
                *
                * This little procedure will attach a device
                * and report the buffer area.
                *
                                    nam     Attach
                                    ttl     Attaches a Device(s)
                *
                * Between IFP1 and ENDC are
                * use /r/defs/os9defs
                * use /r/defs/scfdefs
                *
                                    IFP1
                                    ENDC
                *
       0011               TYPE      set     OBJCT+PRGRM
       0081               REVS      set     REENT+1
                *
       0000 87CD002F                mod     ATTEND,NAME,TYPE,REVS,START,MEMEND
                *
                * Start of data memory
     D 0000                         org     0
     D 0000                         rmb     200         Stack Area
     D 00C8                         rmb     200         Parameter Area
     D 0190               MEMEND    equ     .
                *
       000D 41747461      NAME      fcs     /Attach/
       0013 01            EDITION   fcb     1
                *
       0014               START     equ     *
       0014 A684                    lda     ,x          Check parameter area.
       0016 810D                    cmpa    #$0D        Is it a carriage return?
       0018 270E                    beq     Done        Yes, then we're done!
                * Get a legitimate name
       001A 103F10                  os9     F$PrsNam    Otherwise, get a name
       001D 250A                    bcs     Error       Take care of an error
                * Attach a device
       001F 8600                    lda     #0          Use device capabilities
       0021 103F80                  os9     I$Attach    Attach the device
       0024 2503                    bcs     Error       Take care of an error
       0026 20EC                    bra     START       Go get another device
                *
       0028 5F            Done      clrb                Remove any possible errors
       0029 103F06        Error     os9     F$Exit      Return with error status
       002C C6E003                  emod
                *
                *
       002F               ATTEND    equ     *
                                    end

00000 error(s)
00000 warning(s)
$002F 00047 program bytes generated
$0190 00400 data bytes allocated
$233E 09022 bytes used for symbols
```

SYSGEN

For those who like life as simple a possible, I present to you this month, SYSGEN. It is written in Basic09. I wrote it as one procedure so that the module directory would not be cluttered with many modules. It will create a directory called, "BOOTMODULES" and a file called, "modules". The program goes through the module directory letting you SAVE what you want. The module is saved to the directory and its name added to the list. You can also copy other modules to the directory. And finally, it will even run OS9GEN for you.

The program works using OS9GEN as outlined before. OS9GEN reads 'modlist' and adds modules from BOOTMODULES to the OS9Boot. To create a module list, they must be save from memory. A pointer to the module directory is found in the first page of memory. This page is the system's Direct Page Variables. D.ModDir is at $0026. The first two bytes point the start of the module directory. The next two bytes point to its end. This program was written for level 1. In Level 1, a module entry consists of 4 bytes. The first two point to its location in memory. The third is the link count and the fourth is just there. Level 2, uses a different method for the module directory. There are 8 bytes that describe an entry. The first 2 bytes are for DAT pointer, 2 for the address space size, 2 for the offset to the module and a final 2 for the link count. Level 2 users have a system call, F$GModDr. It copies the module directory to a 2048 byte area. This routine could copy the module directory to a 2K area and extract the necessary information from it. This would eliminate the use of PEEK's and POKE's. The only draw back I can see is, as I understand it, Level 2 memory allocation is dynamic and things aren't always in the same location. The directory entries and DAT images can move. At best I can say, Good Luck. If anyone comes up with something, drop me a line.

Another thing, the first 4 entries--INIT, BOOT, OS9P1 and OS9P2 -- are not saved to the BOOTMODULES. It is assumed that they will be on track 34 or in ROM. Again for Level 2 users, the OS9P2 is not in ROM. So, this may also need to be adjusted. As I said earlier, drop me a line if you have an observation. I thing for Level 1 users this should work nicely.

```
PROCEDURE sysgen
DIM name:STRING[64]
DIM answer:STRING[1]
DIM i,index,FSize:INTEGER
DIM MDStart,MDEnd,ModLoc:INTEGER
DIM offset,location,place:INTEGER
DIM more:BOOLEAN
DIM Pname,Mname:STRING[64]
DIM Defdrive:STRING[10]
DIM Mpath:INTEGER

REM Introduction
PRINT  \ PRINT  \ PRINT
PRINT "OS-9 SYSGEN"
PRINT "by Ron Voigts"
PRINT "April 1, 1986"
PRINT
```

```
REM Get a few OS-9 modules
SHELL "load os9gen makdir copy save"

REM Get working drive
PRINT
PRINT "Enter working drive: ";
INPUT "",Defdrive
IF Defdrive="" THEN
  Defdrive:="/d0"
  PRINT "Default drive: /d0"
ENDIF

REM Get disk in working drive
PRINT
PRINT "Please, put a Formatted disk in "; Defdrive
INPUT "Press <ENTER> when ready",answer
PRINT

REM change to working disk
CHD Defdrive

REM Open path for module list
CREATE #Mpath,"modules":WRITE


REM Create a temporary directory for boot modules
SHELL "makdir "+Defdrive+"/BOOTMODULES"

REM Get location of module directory
PRINT
PRINT "Modules to be in new 'OS9Boot'"
PRINT
MDStart:=PEEK($26)*256+PEEK($27)
MDEnd:=PEEK($28)*256+PEEK($29)

REM Set ModName index
index:=1

REM Fill the array with module names
FOR i:=MDStart TO MDEnd-1 STEP 4
  ModLoc:=PEEK(i)*256+PEEK(i+1)
  IF ModLoc<>0 THEN
    GOSUB 1000
    IF i>MDStart+12 THEN
      PRINT "> "; name; " <   ";
      INPUT "?  Y/N   ",answer
      IF answer="Y" OR answer="y" THEN
        GOSUB 1100
      ENDIF
    ENDIF
  ENDIF
NEXT i

REM Should we add any modules to the bootlist
GOSUB 1200

REM Kill the modules we don't need anymore
REM and close the 'modules' file
CLOSE #Mpath
```

```
    KILL "copy"
    KILL "makdir"
    KILL "save"

    REM Create a System Disk
    PRINT
    PRINT "SYSGEN a system on "; Defdrive;
    INPUT answer
    IF answer="Y" OR answer="y" THEN
      CHD Defdrive+"/BOOTMODULES"
      CHAIN "OS9GEN "+Defdrive+" <"+Defdrive+"/modules"
    ELSE
      PRINT
      PRINT "Use OS9GEN to create a new system volume"
    ENDIF
    END

1000 REM subroutine to get module name
    offset:=PEEK(ModLoc+4)*256+PEEK(ModLoc+5)
    location:=ModLoc+offset
    place:=ADDR(name)
    index:=0
    more:=TRUE
    WHILE more DO
      char:=PEEK(location+index)
      IF char>$7F THEN
        more:=FALSE
        char:=char-$80
      ENDIF
      POKE place+index,char
      index:=index+1
    ENDWHILE
    IF index<64 THEN
      POKE place+index,$FF
    ENDIF
    RETURN


1100 REM subroutine to add name to boot list
     REM and to add file to directory
     PRINT #Mpath,name
     SHELL "save "+Defdrive+"/BOOTMODULES/"+name+" "+name
     RETURN

1200 REM Add modules to the bootlist
     more:=TRUE
     WHILE more DO
       PRINT
       INPUT "Module to be entered: ",Mname
       IF Mname="" THEN
         more:=FALSE
         PRINT
         PRINT "Module additions aborted"
       ELSE
         INPUT "Enter full path name of source: ",Pname
         IF Defdrive=LEFT$(Pname,LEN(Defdrive)) THEN
           SHELL "copy "+Pname+" "+Defdrive+"/BOOTMODULE /"+Mname+" -s"
         ELSE
           SHELL "copy "+Pname+" "+Defdrive+"/BOOTMODULES/"+Mname
         ENDIF
         PRINT Mname; " Copied"
       ENDIF
       PRINT #Mpath,Mname
     ENDWHILE
     RETURN
```

FORTH

A Tutorial Series

By: R.D. Lurie
9 Linda St.
Leominister, MA 01453

WHAT IS FORTH?

People have asked that question many times, and some in not so flattering a tone of voice, either. Forth has been described as:

1. *A programming language*
2. *An operating system*
3. *A philosophy*
4. *None of the above!*

A serious answer to the question would be that FORTH is a combination of the first three.

FORTH AS A PROGRAMMING LANGUAGE

FORTH has all of the attributes of a strongly structured programming language, but I will not bore you with a repetition of the usual description of the virtues of a structured language. Instead, I would simply like to point out some of the more useful characteristics of FORTH, and, yes, I will mention some of the "bad" points. But first, let me give a general description of how the language works.

Each function or procedure in FORTH is called a "word". Each word is "defined" by an expression which may be empty, or it may contain words previously defined; integers may appear within the definition. The definition has this form:: NAME expression ; The only optional part of the definition is the "expression".

The : and ; tell the compiler where to begin and end the definition which has the name " NAME ". In the latest standard, FORTH-83, and in all of the FORTH Interest Group versions, the name can consist of 1 to 31 ASCII characters, all of them significant, and the name can consist of any combination of printable characters. Some commonly used FORTH words are:*:* , . *? ! @ ; EMIT #* There is no such thing as a reserved word, and a word may be redefined as many times as desired. This can get confusing to the programmer, so it is not encouraged.

As generally implemented, FORTH is organized as a *"threaded"* language. This means that, except for a relatively few, very basic primitive words, which are written in native machine code, all of the remaining words are essentially a list of addresses to which the program counter is set, in turn, until that particular word has been completely executed. An example can help to clarify what I mean: *: AA DUP + CR . ;* The word *AA* is a relatively simple definition which duplicates the top 16-bit number on the Data Stack, adds these two numbers, sends a <CR/LF> to the output, and then sends the sum to the output device.

This word is compiled to: Machine Code Function 82 Name letter count AND $80 41C1AA with last letter AND $8011F5 Process a "colon" definition. 0120 Duplicates the top stack number. 01CF Adds the two top stack numbers 0E76. Sends <CR/LF> to the display device 02C6 Send the top stack number to the display 003C End a "colon" definition 5307 Execution address of the previous word Although this example uses the addresses from "FF9", *Wilson Federici's* version of FORTH-83, the idea is the same for all common versions of FORTH for the 68xx series.

FORTH is a compiled language because the definition must be compiled into the proper series of addresses; but FORTH is an interpreted language because each address in the definition is read and acted upon at run-time. Forth is faster than BASIC because it jumps from execution address to execution address without reading intervening code; but FORTH is slower than pure Assembler language because of the extra steps required to find the proper part of the code to be executed next.

Despite being a structured language, FORTH makes no use of variable typing. As a result, the string *"ABCDEFGH"* can be treated as a string of eight ASCII characters, an array of four 8-bit numbers, an array of two 16-bit numbers, or a 32-bit number, interchangeable, as the need might arise. No error signal will ever show up to interfere with the programmer's right to make an unholy mess, if he isn't paying attention! On the other hand, the compiler will not interfere with creative programming, either. But care should be taken to be sure that the programming doesn't get so creative that it cannot be understood or debugged!

FORTH and the 6809 almost appear to be made for each other,since FORTH makes use of four 16-bit pointers/stacks. These usually are:IPY reg., points to next executable word SPU reg., points to top of Data Stack RPS reg., points to top of Return Stack W X reg., points indirectly to word being executed. The D, X, and CC registers can generally be used freely within a definition without concern, but the other three registers must be stored before use and recovered before exiting from a definition.

FORTH AS AN OPERATING SYSTEM

All versions of FORTH that I have ever heard of contain all of the essential elements of a complete operating system:

1. An editor
2. A compiler
3. A run-time interpreter
4. I/O drivers
5. Mass-storage drivers

The usual FORTH editor is not very fancy, compared to most; however, it is quite well suited to working with the commonly used "screen". The FORTH screen consists of 16 lines of 64 characters (1K bytes), and the line editor is quite comfortable with that amount of text at one time. In fact, this article was written with the editor supplied with "FF9".

There are more sophisticated editors available for FORTH, but this one has the advantage of being a virtual standard. Therefore, it is possible to go from one FORTH system to another, even on a different machine, without having to learn a new set of editing commands.

There are full screen editors written in FORTH for FORTH, and there is a great advantage in that, since any FORTH program can be readily customized. I have written a couple of fullscreen editors for my own peculiar combination of hardware, but I guess that I am reactionary enough to be completely comfortable with either the FORTH or the FLEX line editor.

In any case, the editor is always available in the FORTH system. All you have to do to invoke it is to execute the command " EDITOR ".

The compiler is built into the FORTH system, and acts the same way with definitions entered from the keyboard as it does with definitions read from mass-storage. Furthermore, direct execution of keyboard entries work the same as conventional definitions, so that quick tests of alternate forms of a definition can be made before it is committed to a formal definition. This may appear trivial, *but just try it with C!*

The run-time interpreter is the few bytes of machine code which makes the indirect jumps pointed to by the IP. On the 6809, this can be less than a dozen bytes of code.

The relative slowness of FORTH as compared to Assembler programs can be attributed to these bytes of code. FORTH could take about 34 machine cycles to make an indirect jump, as compared to as little as 12 cycles for a simple subroutine branch and return. This is an insignificant difference for some operations, but significant for other operations, and, of course, the time mounts up for many calls. This is pretty much the same kind of problem encountered when comparing C programs to Assembler programs; having many function calls is costly in time, but they sure make the original programming and later debugging easier. It is quite possible that a FORTH program could run faster than a C program doing the same kind of operations.

FORTH contains the necessary I/O drivers to link directly to the physical devices, or else the system can link I/O through the underlying DOS. For example, "FF9" uses the standard I/O calls of FLEX, but it can link directly to the printer, if desired.

One peculiarity of FORTH terminal I/O is that it does not automatically echo the keyboard input to the display output. This must be done explicitly within the program; and it does lead to some confusion in beginning programmers. However, the advantages to this sort of I/O outweigh the disadvantages.

I listed item #5 as the generic "mass-storage", since tape is the storage medium for many CoCo's. However, the real value of FORTH shines through with multiple disks.

FORTH is really designed to use a disk as virtual memory, without any fancy footwork. The FORTH screen is the normal unit of storage, but any number of records can be put into a screen. The FORTH disk is inherently a random access device, and a DSDD80 disk in FLEX format would have a capacity of 710 screens. Furthermore, with two disks, the second disk is a direct continuation of the first disk. In other words, the program can treat the first disk as starting with screen #0 and the second disk starting with screen #711. Of course, with a hard disk, the number of possible consecutive screens is staggering!

FORTH AS A PROGRAMMING PHILOSOPHY

All programming languages have an underlying philosophy which, intentionally or not, have tended to govern the way that language is used. For example, BASIC was meant to be an easy way to learn programming and an easy way to write functioning programs. Pascal was meant to be a teaching language and forces programs which are somewhat difficult to write, but easy to read. FORTH originated as a language for micro-controllers and the programs tend to be terse and memory-miserly.

It is this history of terse programming style which has given FORTH its reputation as being impossible to read or debug. It is also the reason for some of the cryptic symbols that FORTH critics love to point out and laugh at.

The present tendency in general FORTH programming is to use descriptive names for the definitions and to be generous with comments and explanations. Certainly, for tight, ROM code, the old style of FORTH is still best; but most people are not doing that type of programming. Therefore, we should look forward to seeing more well-documented FORTH programs.

Another legacy of the controller philosophy is the almost exclusive use of integer arithmetic and RPN (Reverse Polish Notation--a compliment, not a slur!). The 32-bit integer math can cover just about all of the common needs, and is blindingly fast when compared to the usual floating-point package. PI, for example, is usually represented as 355/113, which is accurate to more significant figures than is PI in the usual floating-point package.

If you must have floating-point math, then packages are available. In fact, there is at least one on COMPUSERVE for the taking, and I am sure that there are more.

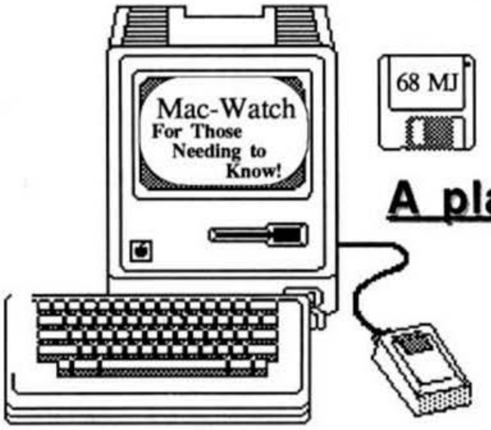
RPN got a bum rap from a lot of people who really did not consider all of its advantages. In fact, the only thing wrong with RPN is that it is not "natural". What they really mean is that they are too lazy to learn a new and better way of doing arithmetic! RPN is easy to learn and use; witness the popularity of HP calcualtors. RPN is the natural math form to use for any stack-related operation, since RPN is basically a Last In-First Out queue. What is more natural for a computer?

In closing, I would like to point out one more "problem" resulting from the FORTH philosophy. Since there was not much RAM left for frills in the early controllers, FORTH does not have a very generous supply of standard warnings and error messages. In other words, FORTH makes very little effort to protect the programmer from himself, as does Pascal, etc., so the programmer must be alert to problems with his own code. On the other hand, FORTH is so easy to troubleshoot, because it is stack oriented and so highly structured, that the error rate tends to be low and easy to recover from.

Certainly, though, the programmer should provide plenty of helpful error messages and graceful recovery routes in all of his applications, no matter what language has been used.

Besides, who could resist a programming language which features the command *" FORGET* ?

+++



The Macintosh™ Section

Reserved as a

A place for your thoughts

Mac-Watch

—

For Those Needing to Know!

BOOKS RECEIVED

an Overlook

- - -

CHILTON'S GUIDE TO MACINTOSH REPAIR and MAINTENANCE

BY: GENE B. WILLIAMS

This title is a **MUST** for the Macintosh user. While it is simplistic in some parts for the more knowledgeable user, it has information that is very valuable to *all* users.

CONTENTS:

1. BEST RESULTS/MINIMAL TIME
2. DIAGNOSIS
3. DISKETTES AND SOFTWARE
4. DISK DRIVES
5. TROUBLESHOOTING THE BOARDS
6. POWER SUPPLIES,KEYBOARD, PRINTERS
7. PERIODIC MAINTENANCE
8. UPGRADING YOUR SYSTEM
9. DEALING WITH TECHNICIANS
10. TROUBLESHOOTING GUIDE

Also several appendixes: from showing the proper way to make a *case popper* (very important, as the case is a *bear to unhinge*, if you don't know how), Taking apart the imagewriter is another 'gotta have' and a fairly complete section on RAM and other upgrades to the Mac.

In our particular case we found the section of 'tables and charts', the most valuable. All in all this book is well worth the cost.

Well illustrated with photos of the various boards, hardware mounting points, methods of removal, and testing procedures. The tables and charts cover system specs, diskette specs, internal drive pin connections, external drive pin connections, power supply pin connections, keyboard pins, the RS-422 serial port pin connections, standard parallel pin connections, standard RS-232 pin connections, IEEE-488 pin allocations, IEEE bus pinouts, Mac-Mac direct connection (5 wire), Mac-Mac direct connection (3 wire), Mac to Modem - 4 wire, audio output jack pins, mouse wiring, self-test sad face icon error codes, self test RAM location chart and self test - OF Exceptions subcodes.

RECOMMENDED

Ed's Note:

This space is reserved for you who have written or called, asking for something on the Mac.

Now to make it work will require your input as well as mine. So if you want the Mac covered in 68 MICRO JOURNAL, then you are going to have to participate!

I will do my part, and hopefully we can make it a worthwhile project. I will expand the space as material is received. Send your material on Mac diskettes in MacWrite and Paint. DMW

UNSQUEEZED
A FLEX Utility

By:
Tom Gilchrist
1450 N. Clarence #108
Wichita, Kansas

If you are using *MODEM9 or MODEM9+* you have probably been searching all those RCPM, RPC, and C-NODE systems for public domain 'C' source code to use under FLEX. If you have had any luck at all you have found some really fine stuff. There are hundreds of solid 'C' programs out there. With the XMODEM file transfer protocol of MODEM9(+) you can download the source error free. However, there is one more unhappy event awaiting you on these systems, the "squeezed" file.

When looking at files on a remote computer you will probably find some that have a "q" in the extension (usually the second letter of the extension). I happened to learn about squeezed files a few months ago.

I had found out about an automatic index program for reports written in 'C'. I found the program, INDEX.CQ, and the documentation, INDEX.DQC, on an RCPM. I listed the files and found they were just what I wanted. Since they were quite long, and the long distance lines were a bit noisy, I used the XMODEM protocol for transfer. Once I listed them on my FLEX system I found out what the term "squeezed" meant. It does not mean a simple space compression scheme (like the one FLEX uses on the disk), it means "compacted beyond recognition".

It seems that the sysops of these RCPM's and etc. wanted to save space on their disks by squeezing files. This also meant that the time of transmission would also be reduced (at 300 baud, long distance, this makes a lot of sense). All you have to do is to unsqueeze files on your computer after transfer. There are unsqueeze (and squeeze for uploading) programs for just about every computer and OS. There are some written in assembler and some written in C. I have been told that the 'C' version was originally written by Dick Greenlaw.

Anyway, I found a version of this unsqueeze program in 'C' for the IBM PC. The 'C' code for PC's is very close to UNIX (and FLEX) 'C'. This version was written by Richard Green and I converted it to INTROL C for FLEX. The program USQ9 will convert squeezed files to ASCII files for FLEX. I have tested the program with files transferred in XMODEM protocol using MODEM9+.

When transferring files, you must use the "b" or binary file mode and not "c" or CP/M mode of MODEM9(+). I name the files on FLEX just as they appear on the RCPM system (INDEX.CQ, INDEX.DQC, etc.). I then use USQ9 to unsqueeze them to INDEX.C, INDEX.DOC, etc. I have found that the ASCII source is 20% to 45% larger.

To compile, I use *INTROL C:*

```
ICC USQ9.C
```

then link:

```
ILINK USQ9 -T=5
```

The resulting program is called USQ9.CMD. The USQ9 program will read a squeezed file and put the ASCII text to stdout. To unsqueeze a file called INDEX.CQ, you type:

```
USQ9 INDEX.CQ
```

The file will be listed on your terminal. If you want to put it into a file you must use the FLEX file re-direction:

```
O 1.INDEX.C USQ9 INDEX.CQ
```

SOME NOTES ABOUT USQ9

1) The original file name is

embedded in the squeeze file and is listed to stdout. You might want to change the program to use this file name for the ASCII text and write the file directly to disk. Be careful if you make this change because the file name embedded might not be in a format for FLEX. An example of one I saw the other day is "CUG.CAT>CUG/CAT.CQT:". I made the program so that you could use multiple file names on the command line. This feature is not important unless you want to make this change.

2) The original program had a dash option to allow you to view the text to stdout as a preview instead of writing to disk. Basically, I changed the program to always preview and force the user re-direct the text to a file.

3) I have found that there is sometimes a few characters of junk at the end of the ASCII text. You can use your editor to delete this junk. I am not sure where the junk is introduced, however, it is a small price to pay.

4) This program should work with a few changes on OS-9, UNIFLEX, and UNIX. (There are versions of usq written for UNIX). Most of the changes will be in the file functions.

5) Just because you transfer 'C' source to FLEX you can't expect to run without checking for compiler differences. CP/M 'C' code (BDS, AZTEC, etc.) will need to be modified to run with INTROL C. The changes are usually not that hard. However, system dependent code will need a lot of work (code that uses CP/M special features and addresses).

Code from IBM PC 'C' compilers will be less work (Lattice, etc.). As with CP/M, code that uses PC BIOS calls will not work without a lot of re-writing.

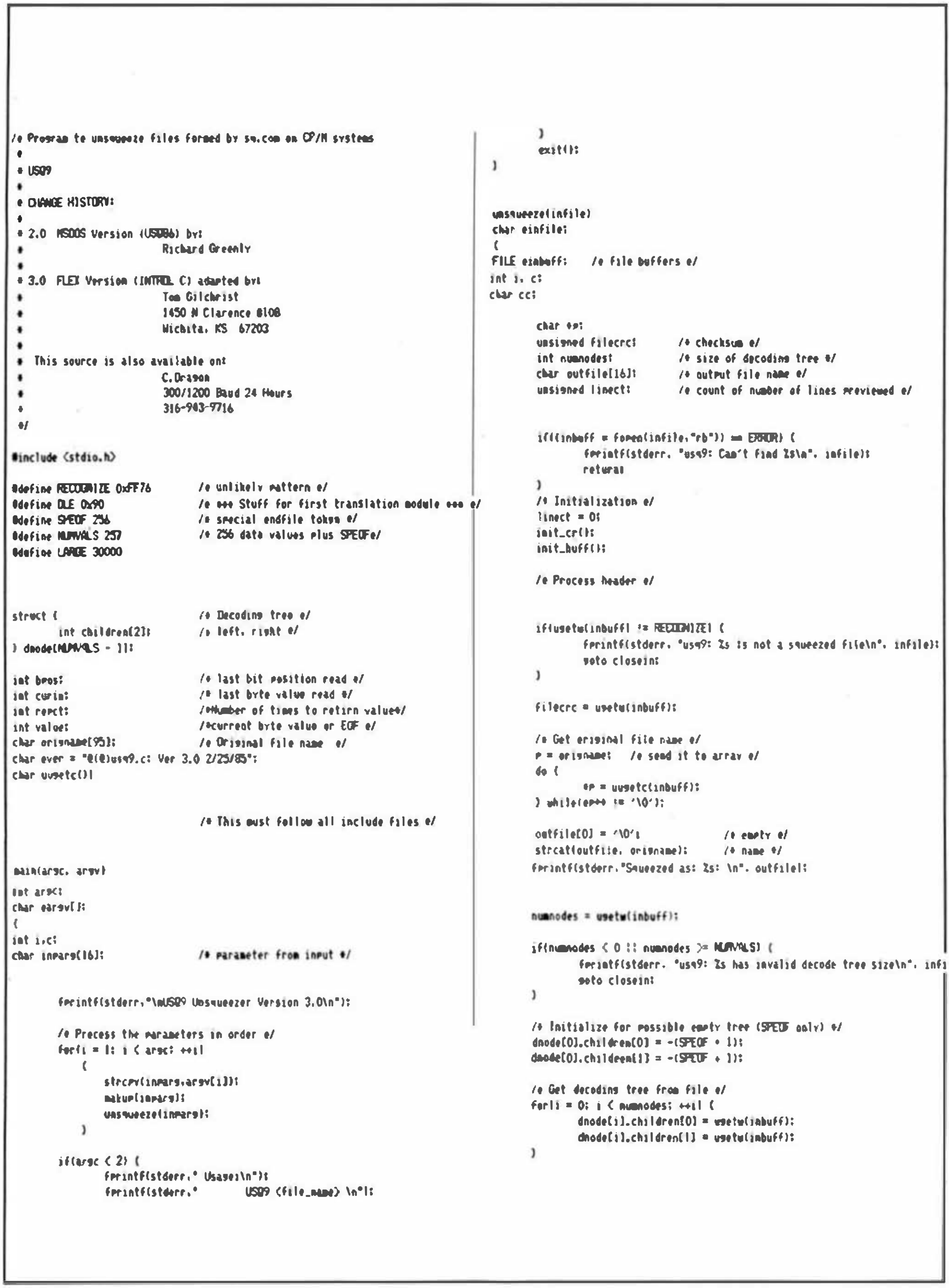
6) Those of you that use COMPUSERVE will not have to use USQ9 in that I have not run across any squeezed file in any SIG.

SOME FINAL THOUGHTS

To find the phone numbers of these remote computers, you will need to do some hunting. There are books that have phone numbers, but these systems change. I have found that the Computer Shopper publication has a good list every month. Almost every month some computer magazine has an article on modem communications with phone numbers of computers. COMPUSERVE has a number of good SIG's including OS-9, PC, CP/M, VAX, etc. in which you can find 'C' code. Of course, you might have some local systems that will not require a long distance call. Check with your local computer stores and computer clubs.

Another good source *of public domain 'C' code is the "C Users Group", PO Box 97, McPherson, KS. 67460.* While this group is primarily interested in BDS C for CP/M, they welcome everyone. There is some neat stuff on the disks they distribute (about $12.00 each). The membership is $10 a year. While they don't distribute on FLEX, you can use the CP/M transfer program published in the May 68 Micro Journal to convert the CP/M disks to FLEX (I understand that the Osborne I, SS/SD disks work on this program, though I have not gotten it to work right on my hardware).

+++



```
/* Program to unsqueeze files formed by sq.com on CP/M systems
 *
 * USQ9
 *
 * CHANGE HISTORY:
 *
 * 2.0  MSDOS Version (USQ86) by:
 *                      Richard Greenly
 *
 * 3.0  FLEX Version (INTROL C) adapted by:
 *                      Tom Gilchrist
 *                      1450 N Clarence #108
 *                      Wichita, KS  67203
 *
 *  This source is also available on:
 *                      C.Dragon
 *                      300/1200 Baud 24 Hours
 *                      316-943-9716
 */

#include <stdio.h>

#define RECOGNIZE 0xFF76        /* unlikely pattern */
#define DLE 0x90                /* *** Stuff for first translation module *** */
#define SPEOF 256               /* special endfile token */
#define NUMVALS 257             /* 256 data values plus SPEOF*/
#define LARGE 30000



struct {                        /* Decoding tree */
        int children[2];        /* left, right */
} dnode[NUMVALS - 1];

int bpos;                       /* last bit position read */
int curin;                      /* last byte value read */
int repct;                      /*Number of times to retirn value*/
int value;                      /*current byte value or EOF */
char origname[95];              /* Original file name  */
char ever = "@(#)usq9.c: Ver 3.0 2/25/85";
char uugetc()l


                                /* This must follow all include files */



main(argc, argv)
int argc;
char *argv[];
{
int i,c;
char inparg[16];                /* parameter from input */


        fprintf(stderr,"\nUSQ9 Unsqueezer Version 3.0\n");

        /* Process the parameters in order */
        for(i = 1; i < argc; ++i)
            {
                strcpy(inparg,argv[i]);
                makup(inparg);
                unsqueeze(inparg);
            }

        if(argc < 2) {
                fprintf(stderr," Usage:\n");
                fprintf(stderr,"        USQ9 <file_name> \n"l;
        }
        exit(1);
}


unsqueeze(infile)
char *infile;
{
FILE *inbuff;    /* file buffers */
int i, c;
char cc;

        char *p;
        unsigned filecrc;       /* checksum */
        int numnodes;           /* size of decoding tree */
        char outfile[16];       /* output file name */
        unsigned linect;        /* count of number of lines previewed */


        if((inbuff = fopen(infile,"rb")) == ERROR) {
                fprintf(stderr, "usq9: Can't find %s\n", infile);
                return;
        }
        /* Initialization */
        linect = 0;
        init_cr();
        init_huff();

        /* Process header */


        if(ugetw(inbuff) != RECOGNIZE) {
                fprintf(stderr, "usq9: %s is not a squeezed file\n", infile);
                goto closein;
        }

        filecrc = ugetw(inbuff);

        /* Get original file name */
        p = origname;   /* send it to array */
        do {
                *p = uugetc(inbuff);
        } while(*p++ != '\0');

        outfile[0] = '\0';                /* empty */
        strcat(outfile, origname);        /* name */
       fprintf(stderr,"Squeezed as: %s: \n", outfile);


       numnodes = ugetw(inbuff);

       if(numnodes < 0 || numnodes >= NUMVALS) {
               fprintf(stderr, "usq9: %s has invalid decode tree size\n", infile);
               goto closein;
       }

       /* Initialize for possible empty tree (SPEOF only) */
       dnode[0].children[0] = -(SPEOF + 1);
       dnode[0].childeen[1] = -(SPEOF + 1);

       /* Get decoding tree from file */
       for(i = 0; i < numnodes; ++i) {
               dnode[i].children[0] = ugetw(inbuff);
               dnode[i].children[1] = ugetw(inbuff);
       }
```

```
        /* Use standard output */
        putchar('\n');
        while((c = usetcr(inbuff)) != EOF) {
                cc = 0x7f & c;  /* strip parity */
                if((cc < ' ') || (cc > '~'))
                        /* Unprintable */
                        switch(cc) {
                        case '\n':      /* newline */
                                ++linect;
                                break;
                        case '\t':      /* tab */
                                break;
                        default:
                                goto next;

                        }
                putchar(cc);
            next: ;
        }
:losein:
        fclose(inbuff);
        return;
}

/* initialize decoding functions */

init_cr()
{
        repct = 0;
}

init_huff()
{
        bpos = 99;      /* force initial read */
}

/* Get bytes with decoding - this decodes repetition,
 * calls getuhuff to decode file stream into byte
 * level code with only repetition encoding.
 *
 * The code is simple passing through of bytes except
 * that DLE is encoded as DLE-zero and other values
 * repeated more than twice are encoded as value-DLE-count.
 */

int usetcr(ib)
FILE *ib;
{
        int c;

        if(repct > 0) {
                /* Expanding a repeated char */
                --repct;
                return(value);
        } else {
                /* Nothing unusual */
                if((c = getuhuff(ib)) != DLE) {
                        /* It's not the special delimiter */
                        value = c;
                        if(value == EOF)
                                repct = LARGE;
                        return(value);
                } else {
                        /* Special token */
                        if((repct = getuhuff(ib)) == 0)
                                /* DLE, zero represents DLE */
                                return (DLE);
                        else {
                                /* Begin expanding repetition */
                                repct -= 2;     /* 2nd time */
                                return (value);
                        }
                }
        }
}

/* Decode file stream into a byte level code with only
 * repetition encoding remaining.
 */

int
getuhuff(ib)
FILE *ib;
{
int i;

        /* Follow bit stream in tree to a leaf*/
        i = 0;  /* Start at root of tree */
        do {
                if(++bpos > 7) {
                        if((curin = uusetc(ib)) == ERROR)
                                return (ERROR);
                        bpos = 0;
                        /* move a level deeper in tree */
                        i = dnode[i].children[1 & curin];
                } else
                        i = dnode[i].children[1 & (curin >>= 1)];
        } while(i >= 0);

        /* Decode fake node index to original data value */
        i = -(i + 1);
        /* Decode special endfile token to normal EOF */
        i = (i == SPEOF) ? EOF : i;
        return (i);
}
/* Various useful things for USQ. */

char uusetc(file)
FILE *file)
 char c;
        c = getc(file);
        return c]


 }

 /* getw() function and transpose */
 int usetw(file)
 FILE *file;
 {
 int cl,ch;
 int cu;
         cl= uusetc(file);
         ch= uusetc(file);
         cu = ((ch << 8) | cl);
         return(cu);
 }

 /* Make a string upper case */

 makup(s)
 char *s;
 {
     while(*s = toupper(*s))
         *s++;
 }
```

☎ Telex 6106006630
(615)842-4600

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601

CoCo OS-9™ FLEX™
SOFTWARE

SPECIAL

K-BASIC

K-BASIC under OS-9 and FLEX will compile TSC BASIC, XBASIC and XPC Source Code Files.

$AVE
$100

K-BASIC now makes the multitude of TSC XBASIC Software available for use under OS-9. Transfer your favorite BASIC Programs to OS-9, compile them, Assemble them, and BINGO -- useable, multi-precision, familiar Software is running under favorite Operating System!

!!! SPECIAL ~~$195.00~~ $99.00 !!!

SCULPTOR

Full OEM & Dealer Discounts Available!

THE SCULPTOR SYSTEM

Sculptor combines a powerful fourth generation language with an efficient database management system. Programmers currently using traditional languages such as Basic and Cobol will be amazed at what Sculptor does to their productivity. With Sculptor you'll find that what used to take a week can be achieved in just a few hours.

AN ESTABLISHED LEADER

Sculptor was developed by professionals who needed a software development tool with capabilities that were not available in the software market. It was launched in 1981 and since then, with feedback from an ever-increasing customer base, Sculptor has been refined and enhanced to become one of the most adaptable, fast, and above all reliable systems on the market today.

SYSTEM INDEPENDENCE

Sculptor is available on many different machines and for most operating systems, including MS DOS, Unix/Xenix and VMS. The extensive list of supported hardware ranges from small personal computers, through multi-user micros up to large minis and mainframes. Sculptor is constantly being ported to new systems.

APPLICATION PORTABILITY

Mobility of software between different environments is one of Sculptor's major advantages. You can develop applications on a stand-alone PC and – without any alterations to the programs – run them on a large multi-user system. For software writers this means that their products can reach a wider marketplace than ever before. It is this system portability, together with high-speed development, that makes Sculptor so appealing to value added resellers, hardware manufacturers and software developers of all kinds.

SPEED AND EFFICIENCY

Sculptor uses a fast and proven indexing technique which provides instant retrieval of data from even the largest of files. Sculptor's fourth generation language is compiled to a compact intermediate code which executes with impressive speed.

INTERNATIONALLY ACCEPTED

By using a simple configuration utility, Sculptor can present information in the language and format that you require. This makes it an ideal product for software development almost anywhere in the world. Australasia, the Americas and Europe – Sculptor is already at work in over 20 countries.

THE PACKAGE

With every development system you receive:

□ A manual that makes sense
□ A periodic newsletter
□ Screen form language
□ Report generator
□ Menu system
□ Query facility
□ Set of utility programs
□ Sample programs

For resale products, the run-time system is available at a nominal cost.

Facts ▪▪▪▪▪ Features ▪▪▪▪▪▪▪▪

DATA DICTIONARY

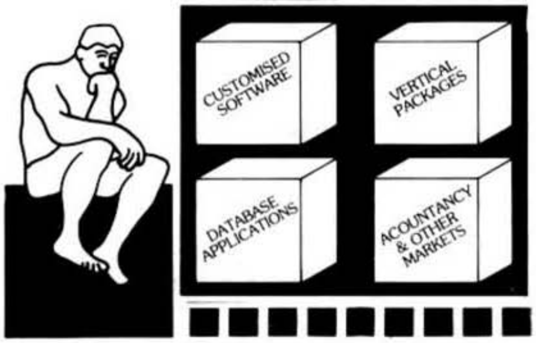
Each file may have one or more record types described. Fields may have a name, heading, type, size, format and validation list. Field type may be chosen from:

□ alphanumeric
□ integer
□ floating point
□ money
□ date

DATA FILE STRUCTURE

□ Packed, fixed-length records
□ Money stored in lower currency unit
□ Dates stored as integer day numbers

**Sculptor for 68020
OS-9 & UniFLEX
$995**



INDEXING TECHNIQUE

Sculptor maintains a B-tree index for each data file. Program logic allows any numbers of alternative indexes to be coded into one other file.

INPUT DATA VALIDATION

Input data may be validated at three levels:

□ automatic by field type
□ validation list in data dictionary
□ programmer coded logic

ARITHMETIC OPERATORS

–	Unary minus
*	Multiplication
/	Division
%	Remainder
+	Addition
–	Subtraction

MAXIMA AND MINIMA

Minimum key length	1 byte
Maximum key length	160 bytes
Minimum record length	3 bytes
Maximum record length	32767 bytes
Maximum fields per record	32767
Maximum records per file	16 million
Maximum files per program	16
Maximum open files	Operating system limit

PROGRAMS

□ Define record layout
□ Create new indexed file
□ Generate standard screen-form program
□ Generate standard report program
□ Compile screen-form program
□ Compile report program
□ Screen-form program interpreter
□ Report program interpreter
□ Menu interpreter

RELATIONAL OPERATORS

=	Equal to
<	Less than
>	Greater than
< =	Less than or equal to
> =	Greater than or equal to
< >	Not equal to
and	Logical and
or	Logical or
ct	Contains
bw	Begins with

SPECIAL FEATURES

□ Full date arithmetic
□ Echo suppression for passwords
□ Terminal and printer independence
□ Parameter passing to sub-programs
□ User definable date format

□ Query facility
□ Reformat file
□ Check file integrity
□ Rebuild index
□ Alter language and date format
□ Setup terminal characteristics
□ Setup printer characteristics

SCREEN-FORM LANGUAGE

□ Programmer defined options and logic
□ Multiple files open in one program
□ Default or programmer processing of exception conditions
□ Powerful verbs for input, display and file access
□ Simultaneous display of multiple records
□ Facility to call sub programs and operating system commands
□ Conditional statements
□ Subroutines
□ Independence of terminal type

- Full Developement Package
•• Run Time Only
••• C Key File Library

OS-9/UnIFLEX
IBM PC Zenix
MS DOS Network -- $995 / $199 / $498
• • •••

68000 UnIFLEX
Altos Zenix
UNIX -- $1595 / $319 / $798
• •• •••

MS DOS
PC DOS -- $595 / $119 / $595
• •• •••

MUSTANG-020™ Users - ask for special discount.

Sculptor is a Trademark of Microprocessor Developments Ltd.

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends—

F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

*OS-9 is a Trademark of Microware and Motorola
*FLEX is a Trademark of Technical Systems Consultants

SOUTH EAST MEDIA
5900 Cassandra Smith Rd
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

** Shipping **
Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA

ASSEMBLERS

ASTRUK09 from S.E. Media – A "Structured Assembler for the 6809" which requires the TSC Macro Assembler.

F, CCF - $99.95

Macro Assembler for TSC – The FLEX STANDARD Assembler.

Special – CCF $35.00; F $50.00

OSM Extended 6809 Macro Assembler from Lloyd I/O. – Provides local labels, Motorola S-records, and Intel Hex records; XREF. GeneÔrate OS-9 Memory modules under FLEX.

FLEX, CCF, OS-9 $99.00

Relocating Assembler/Linking Loader from TSC. -- Use with many of the C and Pascal Compilers.

F, CCF $150.00

MACE, by Graham Trott from Windrush Micro Systems -- Co-Resident Editor and Assembler; fast interactive A.I.. Programming for small to medium-sized Programs.

F, CCF - $75.00

XMACE – MACE w/Cross Assembler for 6800/1/2/3/8 *F, CCF - $98.00*

CROSS-ASSEMBLERS

TRUE CROSS ASSEMBLERS from Computer Systems Consultants - - Supports 1802/5, Z-80, 6800/1/2/3/8/11/HC11, 6804, 6805/HC05/ 146805, 6809/00/01, 6502 family, 8080/5, 8020/1/2/35/C35/39/ 40/48/C48/49/C49/50/8748/49, 8031/51/8751, and 68000 Systems. Assembler and Listing formats same as target CPU's format. Produces machine independent Motorola S-Text.

FLEX, CCF, OS-9, UniFLEX each - $50.00
any 3 - $100.00 - the complete set w/C Source
except the 68000 Source - $200.00
UniFLEX 68000 - $50.00

XASM Cross Assemblers for FLEX from S.E. MEDIA – This set of 6800/1/2/3/5/8, 6301, 6502, 8080/5, and Z80 Cross Assemblers uses the familiar TSC Macro Assembler Command Line and Source Code format. Assembler options, etc., in providing code for the target CPU's.

Complete set, FLEX only - $150.00

CRASMB from LLOYD I/O -- 8-Bit Macro Cross Assembler with same features as OSM; cross-assemble to 6800/1/2/3/4/5/8/9/11, 6502, 1802, 8048 Sers, 80/85, Z-8, Z-80, TMS-7000 sers. Supports the target chip's standard mnemonics and addressing modes.

FLEX, CCF, OS-9 Full package – $399.00

CRASMB 16.32 from LLOYD I/O – Cross Assembler for the 68000.

FLEX, CCF, OS-9 $249.00

UTILITIES

Basic09 XRef from S.E. Media – This Basic09 Cross Reference Utility is a Basic09 Program which will produce a "pretty printed" listing with each line numbered, followed by a complete cross referenced listing of all variables, external procedures, and line numbers called. Also includes a Program List Utility which outputs a fast "pretty printed" listing with line numbers. Requires Basic09 or RunB.

O & CCO obj. only – $39.95; w/ Source - $79.95

Lucidata PASCAL UTILITIES (Requires LUCIDATA Pascal ver 3)

XREF -- produce a Cross Reference Listing of any text; oriented to Pascal Source.

INCLUDE -- Include other Files in a Source Text, including Binary - unlimited nesting.

PROFILER – provides an Indented, Numbered, "Structogram" of a Pascal Source Text File; view the overall structure of large programs, program integrity, etc. Supplied in Pascal Source Code; requires compilation.

F, CCF—EACH 5"- $40.00, 8"- $50.00

DUB from S.E. Media – A UniFLEX BASIC decompiler Re-Create a Source Listing from UniFLEX Compiled basic Programs. Works w/ ALL Versions of 6809 UniFLEX basic.

U - $219.95

☎ (615)842-4600 Telex 5106006630

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601

CoCo OS-9™ FLEX™
SOFTWARE

LOW COST PROGRAM KITS from S.E. Media – The following programs are available for FLEX on either 5 or 8 inch disk.

1. **BASIC TOOL-CHEST $29.95**
   BLISTER.CMD: pretty printer
   LINEXREF.BAS: line cross-referencer
   REMPAC.BAS, SPCPAC.BAS, COMPAC.BAS: remove superfluous code
   STRIP.BAS: superfluous line-numbers stripper
2. **FLEX UTILITIES KIT $39.95**
   CATS.CMD: alphabetically-sorted directory listing
   CATD.CMD: date-sorted directory listing
   COPYSORT.CMD: file copy, alphabetically
   COPYDATE.CMD: file copy, by date-order
   FILEDATE.CMD: change file creation date
   INFO.CMD (& INFOGMX.CMD): tells disk attributes & contents
   RELINK.CMD (& RELINK82): re-orders fragmented free chain
   RESO.CMD: undeletes (recovers) a deleted file
   SECTORS.CMD: show sector order in free chain
   XL.CMD: super text lister
3. **ASSEMBLERS/DISASSEMBLERS UTILITIES $39.95**
   LINEFEED.CMD: 'modularise' disassembler output
   MATH.CMD: decimal, hex, binary, octal conversions & tables
   SKIP.CMD: column stripper
4. **WORD - PROCESSOR SUPPORT UTILITIES $49.95**
   FULLSTOP.CMD: checks for capitalization where required
   BSTYCIT.BAS (.BAC): Stylo to dot-matrix printer program
   NECPRINT.CMD: Stylo to dot-matrix printer filter code
5. **UTILITIES FOR INDEXING $49.95**
   MENU.BAS: selects required program from list below
   INDEX.BAC: word index
   PHRASES.BAC: phrase index
   CONTENT.BAC: table of contents
   INDXSORT.BAC: fast alphabetic sort routine
   FORMATER.BAC: produces a 2-column formatted index
   APPEND.BAC: append any number of files
   CHAR.BIN: line reader

FULL SCREEN FORMS DISPLAY from Computer Systems Consultants -- TSC Extended BASIC program supports any Serial Terminal with Cursor Control or Memory-Mapped Video Displays; substantially extends the capabilities of the Program Designer by providing a table-driven method of describing and using Full Screen Displays.

F and CCF, U - $25.00, w/ Source - $50.00

SOLVE from S.E. Media - OS-9 Levels I and II only. A Symbolic Object/Logic Verification & Examine debugger. Including inline debugging, disassemble and assemble. SOLVE IS THE MOST COMPLETE DEBUGGER we have seen for the 6809 OS-9 series! SOLVE does It all! With a rich selection of monitor, assembler, disassembler, environmental, execution and

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends–

F – FLEX, CCF – Color Computer FLEX
O – OS-9, CCO – Color Computer OS-9
U – UniFLEX
CCD – Color Computer Disk
CCT – Color Computer Tape

*OS-9 is a Trademark of Microware and Motorola
*FLEX is a Trademark of Technical Systems Consultants

SOUTH EAST MEDIA
5900 Cassandra Smith Rd.
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

** Shipping **

Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA

☎ Telex 5106006630
(615)842-4600
South East Media
5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

other miscellaneous commands, SOLVE is the MOST POWERFUL tool-kit item you can own! Yet, SOLVE is simple to use! With complete documentation, a snap! Everyone who has ordered this package has raved! See review - 68 Micro Journal - December 1985. No "blind" debugging here, full screen displays, rich and complete in information presented. Since review in 68 Micro Journal, this is our fastest mover!

Levels I & II only - OS-9 Regular $149.95
SPECIAL INTRODUCTION OFFER *$69.95*

DISK UTILITIES

OS-9 VDisk from S.E. Media – For Level I only. Use the Extended Memory capability of your SWTPC or Gimix CPU card (or similar format DAT) for FAST Program Compiles, CMD execution, high speed inter-process communications (without pipe buffers), etc. - SAVE that System Memory. Virtual Disk size is variable in 4K increÔments up to 960K. Some Assembly Required.

Level I OS-9 obj. $79.95; w/ Source $149.95

O-F from S.E. Media – Written in BASIC09 (with Source), includes: REFORMAT, a BASIC09 Program that reformats a chosen amount of an OS-9 disk to FLEX Format so it can be used normally by FLEX; and FLEX, a BASIC09 Program that does the actual read or write function to the special O-F Transfer Disk; user-friendly menu driven. Read the FLEX Directory, Delete FLEX Files, Copy both directions, etc. FLEX users use the special disk just like any other FLEXdisk

O - 6809/68000 $79.95

LSORT from S.E. Media - A SORT/MERGE package for OS-9 (Level I & II only). Sorts records with fixed lengths or variable lengths. Allows for either ascending or descending sort. Sorting can be done in either ASCII sequence or alternate collating sequence. Right, left or no justification of data fields available. LSORT includes a full set of comments and errors messages.

OS-9 $85.00

HIER from S.E. Media - *HIER is a modern hierarchal storage system for users under FLEX.* It answers the needs of those who have hard disk capabilities on their systems, or many files on one disk - any size. **Using HIER a regular (any) FLEX disk (8 - 5 - hard disk)** can have sub directories. By this method the problems of assigning unique names to files is less burdensome. Different files with the exact same name may be on the same disk, as long as they are in different directories. For the winchester user this becomes a must. Sub-directories are the modern day solution that all current large systems use. **Each directory looks to FLEX like a regular file, except they have the extension '.DIR'.** A full set of directory handling programs are included, making the operation of HIER simple and straightforward. A special install package is included to install HIER to your particular version of FLEX. Some assembly required. Install indicates each byte or reference change needed. Typically - 6 byte changes in source (furnished) and one assembly of HIER is all that is required. No programming required!

***Introduction Special* $69.95**

COPYMULT from S.E. Media – Copy LARGE Disks to several smaller disks. FLEX utilities allow the backup of ANY size disk to any SMALLER size diskettes (Hard Disk to floppies, 8" to 5", etc.) by simply inserting diskettes as requested by COPYMULT. No fooling with directory deletions, etc. COPYMULT.CMD understands normal "copy" syntax and keeps up with files copied by maintaining directories for both host and receiving disk system. Also includes BACKUP.CMD to download any size "random" type file; RESTORE.CMD to restructure copied "random" files for copying, or recopying back to the host system; and FREELINK.CMD as a "bonus" utility that "relinks" the free chain of floppy or hard disk, eliminating fragmentation.

Completely documented Assembly Language Source files included. ALL 4 Programs (FLEX, 8" or 5") $99.50

COPYCAT from Lucidata – *Pascal NOT required.* Allows reading TSC Mini-FLEX, SSB DOS68, and Digital Research CP/M Disks while operating under FLEX 1.0, FLEX 2.0, or FLEX 9.0 with 6800 or 6809 Systems. COPYCAT will not perform miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Also includes some Utilities to help out. Programs supplied in Modular Source Code (Assembly Language) to help solve unusual problems.

F and CCF 5" - $50.00 F 8" - $65.00

FLEX DISK UTILITIES from Computer Systems Consultants – Eight (8) different Assembly Language (w/ Source Code) FLEX Utilities for every FLEX Users Toolbox: Copy a File with CRC Errors; Test Disk for errors; Compare two Disks; a fast Disk Backup Program; Edit Disk Sectors; Linearize Free-Chain on the Disk; print Disk Identification; and Sort and Replace the Disk Directory (in sorted order). – PLUS - - Ten XBASIC Programs including: A BASIC Resequencer with EXTRAs over "RENUM" like check for missing label definitions, processes Disk to Disk instead of in Memory, etc. Other programs Compare, Merge, or Generate Updates between two BASIC Programs, check BASIC Sequence Numbers, compare two unsequenced files, and 5 Programs for establishing a Master Directory of several Disks, and sorting, selecting, updating, and printing paginated listings of these files. A BASIC Cross-Reference Program, written in Assembly Language, which provides an X-Ref Listing of the Variables and Reserved Words in TSC BASIC, XBASIC, and PRECOMPILER BASIC Programs.

ALL Utilities include Source2 (either BASIC or A.L. Source Code).
F and CCF - $50.00
BASIC Utilities ONLY for UniFLEX -- $30.00

COMMUNICATIONS

CMODEM Telecommunications Program from Computer Systems Consultants, Inc. – Menu-Driven; supports Dumb-Terminal Mode, Upload and Download in non-protocol mode, and the CP/M "Modem7" Christensen protocol mode to enable communication capabilities for almost any requirement. Written in "C".

FLEX, CCF, OS-9, UniFLEX; with complete Source $100.00 without Source $50.00
UniFLEX 68000 with complete Source $100.00

X-TALK from S.E. Media - X-TALK consists of two disks and a special cable, the hookup enables a 6809 SWTPC computer to dump UniFLEX files directly to the UniFLEX MUSTANG-020. This is the ONLY currently available method to transfer SWTPC 6809 UniFLEX files to a 68000 UniFLEX system. Gimix 6809 users may dump a 6809 UniFLEX file to a 6809 UniFLEX five inch disk and it is readable by the MUSTANG-020. The cable is specially prepared with internal connections to match the non-standard SWTPC SO/9 I/O Db25 connectors. A special SWTPC S+ cable set is also available. Users should specify which SWTPC system he/she wishes to communicate with the MUSTANG-020. The X-TALK software is furnished on two disks. One eight inch disk contains S.E. Media modem program C-MODEM (6809) and the other disk is a MUSTANG-020 five inch disk

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends–

F – FLEX, CCF – Color Computer FLEX
O – OS-9, CCO – Color Computer OS-9
U – UniFLEX
CCD – Color Computer Disk
CCT – Color Computer Tape

*OS-9 is a Trademark of Microware and Motorola
*FLEX is a Trademark of Technical Systems Consultants

South East Media
5900 Cassandra Smith Rd.
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

** Shipping **

Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA

with C-MODEM (68020). Text and binary files may be directly transferred between the two systems. The C-MODEM programs are unaltered and perform as excellent modem programs also. X-TALK can be purchased with or without the special cables, but this special price is available to registered MUSTANG-020 users only.

X-TALK Complete (cable, 2 disks) $99.95
X-TALK Software (2 disks only) $69.95
X-TALK with CMODEM Source $149.95

XDATA from S.E. Media - A COMMUNICATION Package for the UniFLEX Operating System. Use with CP/M, Main Frames, other UniFLEX Systems, etc. Verifies Transmission using checksum or CRC; Re-Transmits bad blocks, etc.

U - $299.99

EDITORS & WORD PROCESSING

JUST from S.E. Media – Text Formatter developed by Ron Anderson; for Dot Matrix Printers, provides many unique features. Output "Formatted" Text to the Display. Use the FPRINT.CMD supplied for producing multiple copies of the "Formatted" Text on the Printer INCLUDING IMBEDDED PRINTER COMMANDS (very useful at other times also, and worth the price of the program by itself). "User Configurable" for adapting to other Printers (comes set up for Epson MX-80 with Grafrax); up to ten (10) imbedded "Printer Control Commands". Compensates for a "Double Width" printed line. Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc. Use with PAT or any other editor.

* Now supplied as a two disk set;
Disk #1: JUST2.CMD object file, JUST2.TXT PL9 source: FLEX - CC
Disk #2: JUSTSC object and source in C: FLEX - OS9 - CC

The JTSC and regular JUST C source are two separate programs. JTSC compiles to a version that expects TSC Word Processor type commands, (.pp .sp .ce etc.) Great for your older text files. The C source compiles to a standard syntax JUST.CMD object file. Using JUST syntax (.p .u .y etc.) With all JUST functions plus several additional printer formatting functions. Reference the JUSTSC C source. For those wanting an excellent BUDGET PRICED word processor, with features none of the others have. This is it!

Disk (1) - PL9 FLEX only - F & CCF - $49.95
Disk Set (2) - F & CCF & OS9 (C version) - $69.95
OS-9 68K000 complete with Source - $79.95

PAT from S.E. Media - A full feature screen oriented TEXT EDITOR with all the best of "PIE™". For those who swore by and loved only PIE, this is for you! All PIE features and much more! Too many features to list. And if you don't like these, change or add your own. PL-9 source furnished. "C" source available soon. Easily configured to your CRT, with special config section.

Regular FLEX $129.50
' SPECIAL INTRODUCTION OFFER ' $79.95
SPECIAL PAT/JUST COMBO (w/source)
FLEX $99.95
OS-9 68K Version $229.00
SPECIAL PAT/JUST COMBO 68K $249.00
Note: *JUST in "C" source available for OS-9*

CEDRIC from S.E. Media - A screen oriented TEXT EDITOR with availability of 'MENU' aid. Macro definitions, configurable 'permanent definable MACROS' - all standard features and the fastest 'global' functions in the west. A simple, automatic terminal config program makes this a real 'no hassel' product. Only 6K in size, leaving the average system over 165 sectors for text buffer - appx. 14,000 plus of free memory! Extra fine for programming as well as text.

Regular $129.95
SPECIAL INTRODUCTION OFFER FLEX $69.95

☎ (615)842-4600
Telex 5106006630

South East Media
5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601

CoCo OS-9™ FLEX™
SOFTWARE

BAS-EDIT from S.E. Media - A TSC BASIC or XBASIC screen editor. Appended to BASIC or XBASIC, BAS-EDIT is transparent to normal BASIC/XBASIC operation. Allows editing while in BASIC/XBASIC. Supports the following functions: OVERLAY, INSERT and DUP LINE. Make editing BASIC/XBASIC programs SIMPLE! A GREAT time and effort saver. Programmers love it! NO more retyping entires lines, etc. Complete with over 25 different CRT terminal configuration overlays.

FLEX, CCF, STAR-DOS Regular $69.95

Limited Special Offer: $39.95

SCREDITOR III from Windrush Micro Systems – Powerful Screen-Oriented Editor/Word Processor. Almost 50 different commands; over 300 pages of Documentation with Tutorial. Features Multi-Column display and editing, "decimal align" columns (AND add them up automatically), multiple keystroke macros, even/odd page headers and footers, imbedded printer control codes, all justifications, "help" support, store common command series on disk, etc. Use supplied "set-ups", or remap the keyboard to your needs. Except for proportional printing, this package will DO IT ALL!

6800 or 6809 FLEX or SSB DOS, OS-9 - $175.00

SPELLB "Computer Dictionary" from S.E. Media – OVER 150,000 words! *Look up a word from within your Editor or Word Processor (with the SPH.CMD Utility which operates in the FLEX UCS).* Or check and update the Text after entry; ADD WORDS to the Dictionary, "Flag" questionable words in the Text, "View a word in context" before changing or ignoring, etc. SPELLB first checks a "Common Word Dictionary", then the normal Dictionary, then a "Personal Word List", and finally, any "Special Word List" you may have specified. SPELLB also allows the use of Small Disk Storage systems.

F and CCF - $129.95

STYLO-GRAPH from Great Plains Computer Co. – A full-screen oriented WORD PROCESSOR – (uses the 51 x 24 Display Screens on CoCo FLEX/STAR-DOS, or PBJ Wordpak). Full screen display and editing; supports the Daisy Wheel proportional printers.

NEW PRICES 6809 CCF and CCO - $99.95,
F or O - $179.95, U - $299.95

STYLO-SPELL from Great Plains Computer Co. – Fast Computer Dictionary. Complements Stylograph.

NEW PRICES 6809 CCF and CCO - $69.95,
F or O - $99.95, U - $149.95

STYLO-MERGE from Great Plains Computer Co. – Merge Mailing List to "Form" Letters, Print multiple Files, etc., through Stylo.

NEW PRICES 6809 CCF and CCO - $59.95,
F or O - $79.95, U - $129.95

STYLO.PAK -- Graph + Spell + Merge Package Deal!!!

F or O - $329.95, U - $549.95
O, 68000 $595.00

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends–

F – FLEX, CCF – Color Computer FLEX
O – OS-9, CCO = Color Computer OS-9
U – UniFLEX
CCD – Color Computer Disk
CCT – Color Computer Tape

* OS-9 is a Trademark of Microware and Motorola
* FLEX is a Trademark of Technical Systems Consultants

South East Media
5900 Cassandra Smith Rd
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

**** Shipping ****

Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA

☎ (615)842-4600 Telex 5106006630

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601

CoCo OS-9™ FLEX™
SOFTWARE

PROGRAMMING LANGUAGES

PL/9 from Windrush Micro Systems -- By Graham Trott. A combination Editor Compiler Debugger. Direct source-to-object compilation delivering fast, compact, re-entrant, ROM-able, PIC. 8 & 16-bit Integers & 6-digit Real numbers for all real-world problems. Direct control over ALL System resources, including interrupts. Comprehensive library support; simple Machine Code interface; step-by-step tracer for instant debugging. 500+ page Manual with tutorial guide.

F, CCF - $198.00

PASC from S.E. Media - A Flex9 Compiler with a definite Pascal "flavor". Anyone with a bit of Pascal experience should be able to begin using PASC to good affect in short order. The PASC package comes complete with three sample programs: ED (a syntax or structure editor), EDITOR (a simple, public domain, screen editor) and CHESS (a simple chess program). The PASC package come complete with source (written in PASC) and documentation.

FLEX $95.00

WHIMSICAL from S.E. MEDIA Now supports *Real Numbers.* "Structured Programming" WITHOUT losing the Speed and Control of Assembly Language! Single-pass Compiler features unified, user-defined I/O; produces ROMable Code; Procedures and Modules (including pre-compiled Modules); many "Types" up to 32 bit Integers, 6-digit Real Numbers, unlimited sized Arrays (vectors only); Interrupt handling; long Variable Names; Variable Initialization; Include directive; Conditional compiling; direct Code insertion; control of the Stack Pointer; etc. Run-Time subroutines inserted as called during compilation. *Normally produces 10% less code than PL/9.*

F and CCF - $195.00

FORTH from Stearns Electronics -- A CoCo FORTH Programming Language. Tailored to the CoCo! Supplied on Tape, transferable to disk. Written in FAST ML. Many CoCo functions (Graphics, Sound, etc.). Includes an Editor, Trace, etc. Provides CPU Carry Flag accessibility, Fast Task Multiplexing, Clean Interrupt Handling, etc. for the "Pro". Excellent "Learning" tool!

Color Computer ONLY - $58.95

KANSAS CITY BASIC from S.E. Media - *Basic for Color Computer OS-9* with many new commands and sub-functions added. A full implementation of the IF-THEN-ELSE logic is included, allowing nesting to 255 levels. Strings are supported and a subset of the usual string functions such as LEFT$, RIGHT$, MID$, STRING$, etc. are included. Variables are dynamically allocated. Also included are additional features such as Peek and Poke. A must for any Color Computer user running OS-9.

CoCo OS-9 $39.95

C Compiler from Windrush Micro Systems by James McCosh. Full C for FLEX except bit-fields, including an Assembler. *Requires the TSC Relocating Assembler if user desires to implement his own Libraries.*

F and CCF - $295.00

C Compiler from Introl -- Full C except Doubles and Bit Fields, streamlined for the 6809. Reliable Compiler; FAST, efficient Code. More UNIX Compatible than most.

FLEX, CCF, OS-9 (Level II ONLY), U - $575.00

PASCAL Compiler from Lucidata -- ISO Based P-Code Compiler. Designed especially for Microcomputer Systems. Allows linkage to Assembler Code for maximum flexibility.

F and CCF 5" - $99.95 F 8" - $99.95

PASCAL Compiler from OmegaSoft (now Certified Software) -- For the *PROFESSIONAL;* ISO Based, Native Code Compiler. Primarily for Real-Time and Process Control applications. Powerful; Flexible. Requires a "Motorola Compatible" Relo. Asmb. and Linking Loader.

F and CCF - $425.00 - One Year Maint. $100.00
OS-9 68000 Version - $900.00

KBASIC - from S.E. MEDIA -- A "Native Code" BASIC Compiler which is now Fully TSC XBASIC compatible. The compiler compiles to Assembly Language Source Code. A NEW, streamlined, Assembler is now included allowing the assembly of LARGE Compiled K-BASIC Programs. Conditional assembly reduces Run-time package.

FLEX, CCF, OS-9 Compiler/Assembler $199.00

CRUNCH COBOL from S.E. MEDIA -- Supports large subset of ANSII Level 1 *COBOL* with many of the useful Level 2 features. Full FLEX File Structures, including Random Files and the ability to process Keyed Files. Segment and link large programs at runtime, or implemented as a set of overlays. The System requires 56K and CAN be run with a single Disk System. *A very popular product.*

FLEX, CCF; Normally $199.00
Special Introductory Price $99.95

GAMES

RAPIER - 6809 Chess Program from S.E. Media -- Requires FLEX and Displays on Any Type Terminal. Features: Four levels of play. Swap side. Point scoring system. Two display boards. Change skill level. Solve Checkmate problems in 1-2-3-4 moves. Make move and swap sides. Play white or black. This is one of the strongest **CHESS programs running on any microcomputer,** *estimated USCF Rating 1600+ (better than most 'club' players at higher levels)*

F and CCF - $79.95

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends--

F - FLEX, CCF - Color Computer FLEX
O - OS-9, CCO - Color Computer OS-9
U - UniFLEX
CCD - Color Computer Disk
CCT - Color Computer Tape

*OS-9 is a Trademark of Microware and Motorola
*FLEX is a Trademark of Technical Systems Consultants

SOUTH EAST MEDIA
5900 Cassandra Smith Rd
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

** Shipping **

Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA

MasterCard

DISASSEMBLERS

SUPER SLEUTH from Computer Systems Consultants Interactive Disassembler; extremely *POWERFUL!* Disk File Binary/ASCII Examine/Change, Absolute or FULL Disassembly. XREF Generator, Label "Name Changer", and Files of "Standard Label Names" for different Operating Systems.

Color Computer SS-50 Bus (all w/ A.L. Source)
CCD (32K Req'd) Obj. Only $49.00
F, $99.00 - CCF, Obj. Only $50.00 U, $100.00
CCF, w/Source $99.00 O, $101.00
CCO, Obj. Only $50.00

DYNAMITE+ -- Excellent standard "Batch Mode" Disassembler. Includes XREF Generator and "Standard Label" Files. Special OS-9 options w/ OS-9 Version.

CCF, Obj. Only $100.00 - CO, Obj. Only $59.95
F, " " $100.00 - O, object only $150.00
U, " " $300.00

DATA-BASE ACCOUNTING

XDMS from Westchester Applied Business Systems - Powerful DBMS; M.L. program will work on a single sided 5" disk, yet is F-A-S-T. XDMS Level I provides an "entry level" System for defining a Data Base, entering and changing the Data, and producing Reports. XDMS Level II adds the POWERFUL "GENERATE" facility with an English Language Command Structure for manipulating the Data to create new file Structures, Sort, Select, Calculate, etc. XDMS Level III adds special "Utilities" which provide additional ease in setting up a Data Base, such as copying old data into new Data Structures, changing System Parameters, etc.

XDMS System Manual - $24.95
XDMS Lvl I - F & CCF - $129.95
XDMS Lvl II - F & CCF - $199.95
XDMS Lvl III - F & CCF - $269.95

XDMS IV from Westchester Applied Business Systems - *XDMS IV* is a brand new approach to data management. It not only permits users to describe, enter and retrieve data, *but also to process entire files producing customized reports, screen displays and file output.* Processing can consist of any of a set of standard high level functions including record and field selection, sorting and aggregation, lookups in other files, special processing of record subsets, custom report formatting, totaling and subtotaling, and presentation of up to three related files as a "database" on user defined output reports.

*XDMS IV - F, CCF STAR-DOS, SK*DOS $350.00*
Upgrades to XDMS IV - $250.00

MISCELLANEOUS

TABULA RASA SPREADSHEET from Computer Systems Consultants -- TABULA RASA is similar to DESKTOP/PLAN; provides use of tabular computation schemes used for analysis of business, sales, and economic conditions. Menu-driven; extensive report-generation capabilities. Requires TSC's Extended BASIC.

F and CCF, U - $50.00, w/ Source - $100.00

DYNACALC -- Electronic Spread Sheet for the 6809 and 68000.

F, OS-9 and SPECIAL CCF - $200.00, U - $395.00

OS-9 68K - $895.00

FULL SCREEN INVENTORY/MRP from Computer Systems Consultants -- Use the Full Screen Inventory System/Materials Requirement Planning for maintaining inventories. Keeps item field file in alphabetical order for easier inquiry. Locate and/or print records matching partial or complete item, description, vendor, or attributes; find backorder or below stock levels. Print-outs in item or vendor order. MRP capability for the maintenance and analysis of Hierarchical assemblies of items in the inventory file. Requires TSC's Extended BASIC.

F and CCF, U - $50.00, w/ Source - $100.00

FULL SCREEN MAILING LIST from Computer Systems Consultants -- The Full Screen Mailing List System provides a means of maintaining simple mailing lists. Locate all records matching on partial or complete name, city, state, zip, or attributes for Listings or Labels, etc. Requires TSC's Extended BASIC.

F and CCF, U - $50.00, w/ Source - $100.00

DIET-TRAC Forecaster from S.E. Media -- An XBASIC program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P G%) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual. Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individual are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. Provides number of days and daily calendar after weight goal and calorie plan is determined.

F - $59.95, U - $89.95

!!! Please Specify Your Operating System & Disk Size !!!

Availability Legends

F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

*OS-9 is a Trademark of Microware and Motorola
*FLEX is a Trademark of Technical Systems Consultants

** Shipping **

Add 2% U.S.A.
(min. $2.50)
Add 5% Surface Foreign
10% Air Foreign

VISA
MasterCard

IMS

Information Management System

From: **CLEARBROOK SOFTWARE GROUP**

Information Management System - featuring both relational and network capabilities. Including tools to get the job done *right*! An OS-9 application.

IMS is designed to operate on any OS-9 system level II.

SPECIFICATIONS:

Maximum data file size	OS limited
Maximum # of records per file	OS limited
Maximum # fields/record	memory limited
Maximum # bytes per record	memory limited
Minimum # bytes per record	5
Maximum # open files	5 or 6
Maximum # keys per field	127
Maximum length of single field	memory limited
Maximum length name (field or variable)	255
Maximum # lines per module	memory limited.

Note:
OS limited: Limited by disk capacity and OS.
Memory limited: OS-9 dependent

SYSTEM REQUIREMENTS

OS-9 level II 6809

RAM: 128K or more

Disk capacity: 2 DS DD of 250K or more per disk. Hard disk recommended for advanced applications.

CRT: Absolute cursor addressing, clear screen, clear to end of line.

Printer: ASCII printer with 80 or more columns, responding to ASCII formfeed.

UPDATE POLICY:

Updates are free for the first year. After one (1) year, updates are $15.00 USA, overseas $20.00.

The above information is extracted directly from the manual. As I have received updates without applying, I can only say that is one of the better policies I have experienced. It needs to be said, *in the review of any product*, that updates should be timely and no hassle! So far that has been the case with this company.

DOCUMENTATION:

The documentation for this product is above average for the S50 bus type software. The manual is broken down into several major sections, with extended tabs on heavy plastic stock. This may not seem like much up front. But if you have ever sat down to work with a new piece of software and had to thumb and shift through a hundred or so pages, looking for one particular item, then you will immediately appreciate the manual. Over 150 pages of tutorial and well laid out logical references. Not to mention the several appendices.

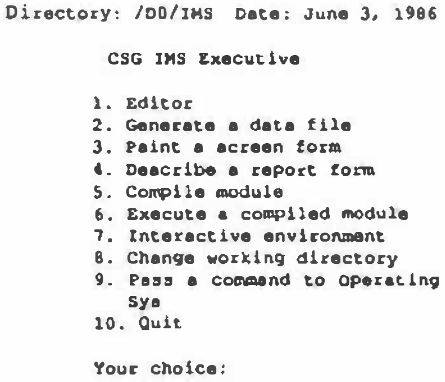
Even the spiral binding is the type that allows for easy insertion of updates, etc. and the disk update we received is enclosed in its own heavy plastic binder.

The only complaint I have with the documentation, even with the section tabs is that there is no *index*. I should hope one will be made available in the next update. One hundred and 50 pages, more or less, is a lot of flipping for the beginner.

INTRODUCTION-TUTORIAL

The first 60 or so pages are devoted to a tutorial, step by step application. It assumes the user is not too well established in developing programs of this caliber. Nice for the beginner and even us *old hackers*. I found it both instructional and beneficial, despite having over 12 years experience developing applications software.

Upon entering into a session by typing **IMS**, OS-9 loads a screen called *'main menu'*. As follows:



```
Directory: /D0/IMS  Date: June 3, 1986

            CSG IMS Executive

            1. Editor
            2. Generate a data file
            3. Paint a screen form
            4. Describe a report form
            5. Compile module
            6. Execute a compiled module
            7. Interactive environment
            8. Change working directory
            9. Pass a command to Operating
               Sys
            10. Quit

            Your choice:
```

FOR THOSE WHO
68 MICRO JOURNAL™
NEED TO KNOW!

```
CSG Information Management System
Version 1.x, Serial number xxxxxx
(c) 1985, Clearbrook Software Group
Inc.
```

You are now working inside IMS and with a structured menu. This is the manner in which most all better development systems operate. Less chance for error and it makes everything compatible. For those developing software for resale, this is a must.

SELECTING

1. Editor (TX)
Immediately displayed are two items. A menu of files available to be worked on - *(.imo - .ide)*. .imo files are IMS modules. .ide are data descriptors. The extensions are recommended for all IMS files.

2. Generate a data file
Prompt is: Name of file descriptor:

You type in the name of a file from the menu list above.

3. Paint a screen form
Prompt is: Data base file(s):

This option allows you to design the file form. Also can generate an IMS module to maintain files used in this form.

4. Describe a report format
Prompt is: Data base file(s):

This option then waits for you to type in the files you wish to create or edit a report form. Available data files are listed above the prompt. Also it allows you to design the report form and generate an IMS module to maintain the report.

5. Compile module
Prompt is: Source file to compile:

Possible file names are listed above the prompt. The module was created in the editor. Then it must be compiled.

6. Execute a compiled module
Prompt is: Module to execute...

Enter a compiled module. It must have been compiled first, using item #5 above.

7. Interactive environment
This menu item allows queries on the file information and gets instant results.

8. Change working directory
Just what it says. Change your working directory to any other directory, on the system.

9. Pass a command to the operating system
Again, just what it says. Execute a shell command.

10. Quit
Not too hard to figure out.

Included with the software package is a set of tutorial and example programs. The one used in the tutorial is a well developed 'maillist & Payroll'.

If it is called in the item #6 of the above menu, a form will be printed to the screen. This particular 'maillist' has 5 pre-defined fields. It is easily remodeled to suit others needs.

NOTE: both Payroll and Maillist can easily be developed into very complete programs. They are very useable as they come. However, knowing many of you, my guess is that some changes will be made.

After the form is filled out, by typing the necessary information as requested, a small menu is printed on the bottom of the screen as follows:

```
Insert Update Clear Delete First
Last Next Previous Key Search
 Quit
```

Each is selected by typing in the first letter of the desired word action.

Insert adds the screen data to the disk file.

Update This updates a file that was previously found using *Search, First, Last, Next or Previous.*

Clear This will clear the fields on the screen. However, the file is untouched unless you *Update* after the *Clear.*

Delete This deletes the record displayed on the screen.

First Selects the first record in the file. This could be displayed as the first alpha sorted key file, if the key exists.

Last This displays the last record, according to the key.

Next According to key, the next record is displayed.

Previous Right, this displays the previous record. See how easy it is getting. All fooling aside, it is an easy system to master, considering its power and rich set of functions.

Key The key option prints the prompt:

```
Choose one field:
  *   1. - NOKEY
      2. - name
Selection?
```

The files are ordered by the current key. The asterisk denotes the key in use. In this case the NOKEY. However, a 'name' key has also been created by alphabetical order. NOKEY is special in that it retrieves data in the order it is created.

EDITORS

There are three different editors that are furnished with the system. First is TX, a very nice text editor. TX is a standalone program, furnished to make editing files of this nature easy in OS-9 *(know what I mean, Vern?)* TX has many of the text editing functions found in more expensive text editors. Of course it, as well as all the other IMS facilities, requires that you have used the accompanying configure facilities to configure your CRT to the (at least) minimum requirements (above).

A 'keyboard' composing the **K J H L** keys control rapid cursor movements. Also there are the usual advanced editing functions, including Help, Cut, Duplicate, Paste, Write, etc.

Screens are accessed by the ^N to next screen and ^P to previous screen. Also included are the usual Find/Replace, Delete Character (under the cursor ^D, delete left of the cursor DEL key). Also ^U Undeletes a character deleted by the DEL key. That character having been stored in a special buffer. This action causes the last deleted character to be placed at the current cursor position. Another ^U would cause the next character in the delete buffer to be placed at the cursor position, etc. OS-9 commands can be passed to the shell from this editor also.

Of course, in addition to these, and many other features too numerous to detail here, are the normal editor functions such as SAVE, LOAD, CLEAR, and QUIT.

FORMS EDITOR

The Forms Editor is an excellent editor for creating and maintaining single screen forms. Forms so created allow interactive I/O with one or more data files.

As with the text editor the keyboard has the same cursor keys setup, with many of the same commands (those necessary in the design of a form).

One function not to be overlooked is the MASK function. MASKs are field names. By the use of the MASK command, general forms may be set up and field names only changed (inserted as a new form is needed). A real time saver and necessary for those who will develop and sell software developed under IMS.

Additionally there are commands for drawing the boxes (prompts, headers, etc.) practically anywhere on the screen, also boxes may be deleted by these functions.

The normal open, close, help, load, etc. commands exist as would be expected.

A special GENERATE command allows the forms editor to generate a program in the applications language, that will use any screen form. It allows date entry, editing and maintenance.

REPORTS EDITOR

This program allows a simple and easy way to define a report form. The user fashions the report form to suit the requirements of the related data files, to suit whatever degree of complexity or simplicity he/she desires. It may be edited and changed at a later date, if desired.

Again the normal keyboard and cursor keys prevail. Most all the normal editing features are included, plus some special ones necessary to generate the desired report forms.

Very complex and complete reports are easily generated by this program. This and the forms generator program can literally save the average programmer hundreds, if not a few thousand hours of programming by most any other method, for projects of larger size. But then I guess that is what advanced development tools such as IMS are all about.

IMS Language

The heart of the system is the IMS applications language, ***IMSI***, with a very rich and complete set of commands, instructions, functions and I/O directives. File structure is a concurrent B+tree type. This is very fast and allows concurrent updates. The nature of IMSI encourages structured programming.

Data types supported are INTEGER, LONG INTEGER, REAL, DATE AND TEXT.

Arrays are multi-dimensional. Date formats are user defined. The system supports up to 161 million records. Which all sums up to a very powerful but easy to use development system!

Conditional/Relational operators: AND, NOT, OR, XOR, relationals: < > = <= >= <> BW CT SL.

Conversions: DATE, INTEGER, LONG, REAL, TEXT, VALUE.

Date related: DATE, TIME, TODAY.

Error trapping: ERROR, RESUME, RESUME AT, RETRY, SET TRAP.

File related: CHD, CHECK, CLEAR, CLOSE, COPY, DELETE, DUPLICATE, EOF, FIELD, field name, FIND, FILE TAG, INSERT, KEY, key clause, LINK, LIST, MARK, MARKED, OPEN, range, RECORD, REINDEX, SCAN, UNLINK, UNMARK, UPDATE, USE.

Input related: ENTER, ESCAPE, GETKEY, INPUT, KEY PRESSED, MASK, SET.

Miscellaneous: arrays, constants, data type, EXECUTE, expression, identifiers, LET, NOTE, operators (+ - * %), SET, SHELL.

Numerical functions: ABS, ASCII, INTEGER, LENGTH, LONG, MAX, MFREE, MIN, REAL, ROUND, SIGN, SQRT, SUBSTR, TRUNCATE, VALUE.

Output related: CLEAR FORM, CLEAR LINE, CLEAR SCREEN, DISPLAY, EJECT PAGE, HELP, LINE NUMBER, LOCATE, MASK, PAGE NUMBER, PRINT, SET, TAB.

Program control: CALL, CASE (WHEN, ENDWHEN,ENDCASE), CHAIN, END, EXIT, GOSUB-RETURN, GOTO, IF (ELSE ENDIF), LABEL, LOOP, END LOOP, MODULE, QUIT, REDO, REPEAT-UNTIL, WHILE-ENDWHILE.

Text functions: CAP$, CHR$, LEFT$, LENGTH, LIBRARY$, MASK, MAX, MID$, MIN, PADCENTER$, PADRIGHT$, RIGHT$, SOUND$, SUBSTR, TEXT, TIME, TRIM$, VALUE.

WHEEEEEEEE!

Now, there is no way I can demonstrate or explain all of the above IMSI functions, etc, in a review. Many of you can readily deduct the meaning. Others may not be so easily guessed, but I can assure you that there is something there for practically any application! However, if you are on the lookout for a really *top class OS-9 development system*, the above should convince you this package deserves your close attention.

Additional features

Some of the additional features are as follows. A Universal Terminal Driver. This is a collection of programs to allow the user total terminal control. Four utilities are related and furnished: *MKTERM*-make terminal drivers. *ASSOC*-associate a terminal to a physical device. *NMALL*-names all terminal drivers and known physical devices associations. *TNAME*-names the terminal driver a device is associated with.

Since starting this review I have received two updates. Currently to version 1.2. In addition to the above which applies to version 1.0 are the following additions. *UNLOCK* <file tag>.TX has been expanded with an '*over write*' function. And the universal terminal driver supports additional addressing types. This information is of more importance to present users, rather than any new users, due to the fact that Clearbrook's policy of updates is very timely.

TX can be ordered as a separate program for the sum of $50.00.

A utility called LIBR (not furnished for this review) is a utility that is for maintaining libraries of ROF object modules. It is useable with Microware C and the RMA assembler. 6809 version $50.00, from Clearbrook, see advertising this issue.

Finally, a version for OS-9 Level I is to be offered soon. Also a 68000 version should be available soon. I have not seen either yet, but will let you know about them as soon as they are released for review. However, I am told that the 68000 version will be English Language Interfaced.

Example:

```
WHO HAD SALES OVER $1000 IN MARCH?
```

Instead of (present query type):

```
LIST ALL FOR SALES (3) > 1000 PRINT NAME
```

Neat huh? Bet the 6809 crowd will clamor for that, and soon. Either way, 6809 or 68000 version, IMS is a *MUST* for those doing serious program development!

They did it again. Just I was wrapping this review up and getting it off to 68 Micro Journal, they up and sent me another update, 1.3. Also other information to pass along.

Version 1.3 is now being shipped, and available for udate to registered owners.

Also they claim it is now 28% faster. It was fast already! Several new functions. One sets single/multi user mode. Single user runs even faster. A SORT statement for data file sorting was added.

They report the 68000 version is ready for shipping (no price mentioned, and we did not receive one yet for review, so will have to tell you later). Source programs are compatible between 6809 and 68000 versions, so they say. That's it folks, a wrap.

+++

BIT-BUCKET

By: All of us..........................

"Contribute Nothing
Expect Nothing!"
DMW 1986

MOTOROLA INC.

Microprocessor Products Group
P.O. Box 3600
Austin, Texas 78764

For further information contact:

EDITORIAL CONTACT:
Mark Vercruysse
512/928-6804

READER CONTACT:
Dean Mosley
512/928-2839

INQUIRY RESPONSE:
R.Q. Green
P.O. Box 52073
Phoenix, AZ 85072

MOTOROLA TECHNOLOGY STRENGTHENS THE MC68020 AS THE INDUSTRY STANDARD

Austin, Texas, June 18, 1986... Since its introduction in June 1984, Motorola's MC68020 has become the 32-bit MPU performance standard. Just look at the list of systems using the MC68020 (see attached figures). These are the companies who have publicly announced using the MC68020. Many more are in the confidential design stage.

To make the MC68020 a success, production capability must be in place. In 1984, Motorola shipped 5,000 MC68020 units. The process was tuned to increase yields in 1985, thus allowing the shipment of over 50,000 MC68020 processors. For 1986, Motorola plans to deliver more than a quarter of a million MC68020's. The 32-bit market is here today and thriving.

In order to supply the production volume required, the technology must be in place. Motorola has been manufacturing standard CMOS for over 10 years. This background gives Motorola the experience to develop High speed CMOS (HCMOS) for VLSI devices like the MC68020. The current HCMOS process is 3 years mature.

The MC68020 is processed with 1.7 micron HCMOS using a unique single metal layer with silicide. THe MC68020 chip layout is not interconnect bound, therefore a single layer metal process with silicide is used versus a two layer metal process. This unique HCMOS silicide process allows the MC68020 to run at the highest clock frequency of 20 Mhz and maintain a manufacturable yield.

The U.S government tri services (Army, Navy, Air Force) have instituted a research program named VHSIC (Very High Speed Integrated Circuits) to develop advanced processes, architectures, and packaging. There are two phases in this program:

Phase 1 - Motorola produced a 1K X 4 Static RAM at 25 Mhz using a 1.25 micron CMOS process. Enhancements to this process are now in progress.

Phase 2 - To develop a 0.5 micron CMOS (now in production) and an advanced bipolar process.

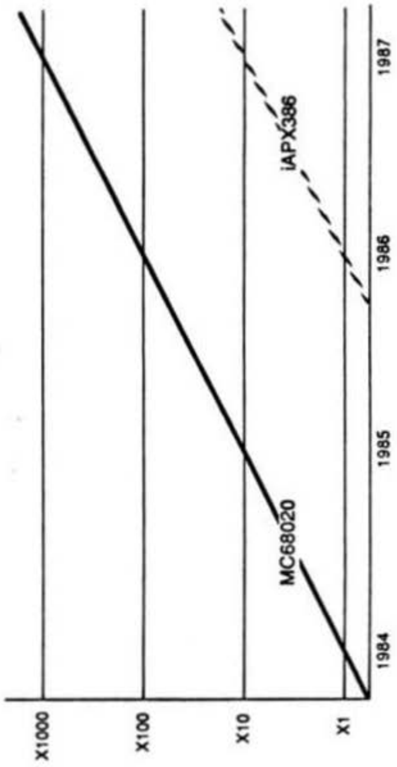
Motorola's standard commercial products like the MC68020 family will utilize this VHSIC expertise in the future to continue offering the highest performance MPU family. The next generation sub-micron HCMOS process for standard commercial products will be in production by 1989.

The Motorola MC68020 has greater than 80% of the 32-bit market. There are several reasons for this success:

1. Highest performance 32-bit MPU at the best price ($174 quantity 100)
2. Complete 32-bit architecture with 17 general purpose registers
3. On-chip instruction cache and pipeline (>33% increase in performance)
4. Parallelism allowing simultaneous instruction and data fetches
5. Highest performance floating point coprocessor (>1 Mega Whetstone) MC68881
6. Versatile Paged Memory Management Unit (PMMU) MC68851

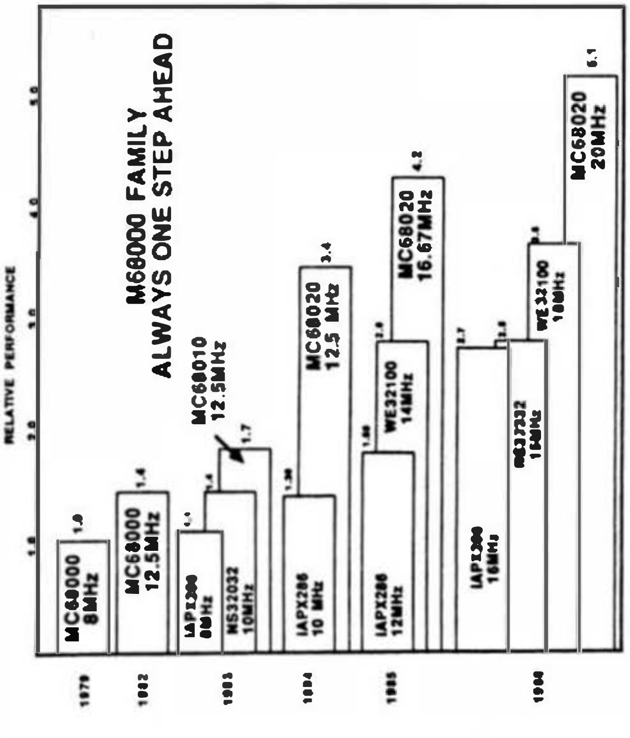
The items listed above are features and capabilities, which are only valuable if they are delivered reliably in high volume at the right price. Motorola is now shipping the MC68020 at 12.5, 16.67, and 20 Mhz in full production.

PRODUCTION LEAD



— 68020 IS OVER 18 MONTHS AHEAD (100X) IN MANUFACTURING

— IT TAKES TIME TO RAMP UP (LOOK AT PAST HISTORY)



MOTOROLA'S VHSIC PROGRAM

PHASE I

- DEVELOPED 1.25 MICRON CMOS
- PRODUCED 1K X 4 SRAM AT 25 MHZ
- ENHANCEMENTS IN PROCESS

PHASE II

- DEVELOPING 0.5 MICRON CMOS AND ADVANCED BIPOLAR TECHNOLOGY

68000 FAMILY PRODUCTS TO UTILIZE VHSIC EXPERTISE

ADVANCED TECHNOLOGY RESEARCH

- VHSIC (VERY HIGH SPEED INTEGRATED CIRCUITS)
- U.S. GOVERNMENT TRI SERVICE PROGRAM
  ARMY, NAVY, AIR FORCE
- TO DEVELOP ADVANCED:
  PROCESSES ARCHITECTURES PACKAGING
- COMPANIES INVOLVED:
  HONEYWELL IBM MOTOROLA TRW
- MOTOROLA: THE ONLY SEMICONDUCTOR MANUFACTURER AWARDED A CONTRACT

MOTOROLA HCMOS TECHNOLOGY

- 3 YEARS MATURE / IN PRODUCTION NOW
- 1.7 MICRON MANUFACTURABLE IN VOLUME TODAY
- SUB 1.0 MICRON BY 1989
- SPEEDS NOW UP TO 20 MHZ
- UNIQUE SINGLE LAYER METAL WITH SILICIDE
  - MANUFACTURABILITY = BETTER YIELDS = LOWER COSTS
  - HIGHEST SPEEDS IN THE INDUSTRY
- HCMOS PRODUCTS INCLUDE:
  - 32 BIT FAMILY (68020 MPU, 68881 FPCP, 68851 PMMU)
  - SERIAL PROCESSING UNITS - SPU (68824 TBC, 68505 XPC)
  - MCU (68HC05C4, 68HC11)
  - STATIC RAMS

PUBLICLY ANNOUNCED MC68020 MPU CUSTOMERS

MULTIUSER OFFICE COMPUTER

- ALPHA MICRO
- ALTOS
- C. ITOH
- CASIO
- CHARLES RIVER DATA
- CONVERGENT TECHNOLOGIES
- DATA-COMP DIVISION/CPI
- DATA MEDIA SYSTEMS
- FORTUNE
- HARRIS
- HONEYWELL
- INTERTECHNIQUE
- MOTOROLA COMPUTER SYSTEMS
- NCR
- PERTEC
- PLEXUS
- QUOTRON
- SPERRY
- TEXAS INSTRUMENTS

PARALLEL PROCESSING

- ARETE
- BOLT BERANEK and NEWMAN
- BURROUGHS
- CALTECH
- FERMI NATIONAL ACCELERATOR LABORATORY
- FERRANTI COMPUTER SYSTEMS
- FLEXIBLE
- ICON
- NIXDORF
- OMNIBYTE
- PARALLEL COMPUTERS
- SANYO
- TANDEM
- WESTINGHOUSE

CAE/GRAPHICS WORKSTATION

- APOLLO
- AT&T
- CADNETICS
- CAMBRIDGE MICROCOMPUTERS
- COMPUTERVISION
- COUNTERPOINT
- CSEE
- HEWLETT PACKARD
- INTEGRATED SOLUTIONS
- KMW SYSTEMS
- MASSCOMP
- PENTAX
- PIXEL SYSTEMS
- SILICON GRAPHICS
- SUMITOMO-DENKO
- SUN
- SUNTEK
- TADPOLE TECHNOLOGY
- TEKTRONIX

ROBOTICS/FACTORY AUTOMATION

- ALLEN-BRADLEY
- AUTOMATIX
- BAILEY CONTROLS
- CIMLINC
- GM
- MITSUBISHI

PBX TELEPHONE SWITCHING

- ALCATEL-THOMSON
- NORTHERN TELECOM
- SIEMENS

VMEbus AND SINGLE BOARD COMPUTERS

- ASEA
- BICC-VERO
- DATA-COMP DIVISION/CPI
- DUAL SYSTEMS
- DY-4 SYSTEMS
- ELECTRONIC MODULAR SYSTEMS
- FORCE COMPUTERS
- GENERAL MICRO SYSTEMS
- GMX
- GOODSPEED SYSTEMS
- HAGENER & KLASSER
- HEURIKON
- IMP
- INTEGRATED SOLUTIONS
- INTELLIGENT SOFTWARE
- IRONICS
- KONTRON
- LYNX BUSINESS MACHINES
- MATROX
- MICROBAR SYSTEMS
- MICROPROJECT
- MIZAR
- MOTOROLA MICROSYSTEMS
- OMNIBYTE
- PACIFIC MICROCOMPUTERS
- PEP
- PERFORMANCE TECHNOLOGIES
- PLESSEY
- SORD COMPUTER
- SYNTEL MICROSYSTEMS
- SYSTEMFORSCHUNG
- VDS
- WESTECH SYSTEMS

DESIGN/DEVELOPMENT SYSTEMS

- APPLIED MICROSYSTEMS
- LANGUAGE RESOURCES
- MICROTEK INTERNATIONAL
- SCIENTIFIC CALCULATION
- TELESIS
- THOMSON LAVAL
- VALID LOGIC

MILITARY

OVER 26 CONFIDENTIAL GOVERNMENT PROJECTS

- AIRCRAFT
- COMMUNICATIONS
- GLOBAL POSITIONING SYSTEMS
- MISSILES
- RADAR
- SHIPS
- SUBMARINES
- TANKS
- AND MORE . . .

MOTOROLA'S 68HC11 DESIGN CONTEST SOLD OUT ANNOUNCES CONTEST II

AUSTIN, TEXAS, JUNE 16, 1986... Motorola Microprocessor Group announces a new Design Contest based on the M68HC11EVB to succeed the original E Squared - MC Contest which has sold out. The M68HC11EVB is an evaluation board for the MC68HC11 single-chip MCU and allows the user to develop software and perform single chip emulation, while helping them become aware of the benefits of EEPROM on MCUs. Contest II features thousands of dollars in prizes and an extended time period to prepare a contest entry.

Two thousand M68HC11EVBs were prepared for the original Design Contest. "We really thought that we had a satisfactory number of boards for the Design Contest", stated Marketing Manager Steve Marsh. "Any time you sell a 2 year supply of product in 2 weeks, you are in for a surprise", he continued.
Due to the unprecedented success of the original M68HC11EVB Design Contest, not all prospective contestants have received contest materials. "Apparently, several Motorola Distributors did not have a system in place to detect that their boards were sold out". Marsh stated. Customers who did not act fast enough to obtain one of the two thousand original contest boards, should order Contest II M68HC11EVBs from their Authorized Motorola Distributor before the July 31, 1986 deadline.

The materials for Contest II and the original Design Contest are virtually the same. Materials contain a M68HC11EVB, Reference Manuals, details for free and discounted software, and much more.

The monitor on the M68HC11EVB features a line assembler/disassembler, a trace feature, multiple breakpoint setting, down load commands, memory and register, displaying and modifying, and a user help command. By connecting the EVB to a target system, such as an IBM PC, the user may emulate the MC68HC11 in the single chip mode of operation. Together the assembler and the EVB provide a very economical means of writing, downloading and debugging user code, and evaluating target system performance. The board may also be used as a stand alone controller, much as in a distributed network processing system.
There will be an unlimited amount of M68HC11EVBs available for Contest II. The EVB is available through Motorola's Authorized Distributors for $168.11. However, Motorola will rebate $68.11 if a qualified design entry is submitted.

For more information contact your local Motorola Sales Office or Authorized Motorola Distributor.

MOTOROLA'S MC68881 FLOATING POINT COPROCESSOR BREAKS THE 1 MILLION WHETSTONE PERFORMANCE BARRIER

Austin, Texas, June 18, 1986... Motorola's MC68881 Floating Point Coprocessor (FPCP), in combination with Motorola's MC68020 32-bit MPU at 16.67 Mhz, has surpassed the 1 million Whetstone performance barrier, as reported by Silicon Valley Software (Fortran compiler), Sun (engineering workstations), and Masscomp (engineering workstations). The MC68881's performance is due to a multitude of optimized floating point routines in hardware coupled with a 16.67 Mhz clock frequency. The Whetstone benchmark is a program designed to test the computational capabilities of a system.

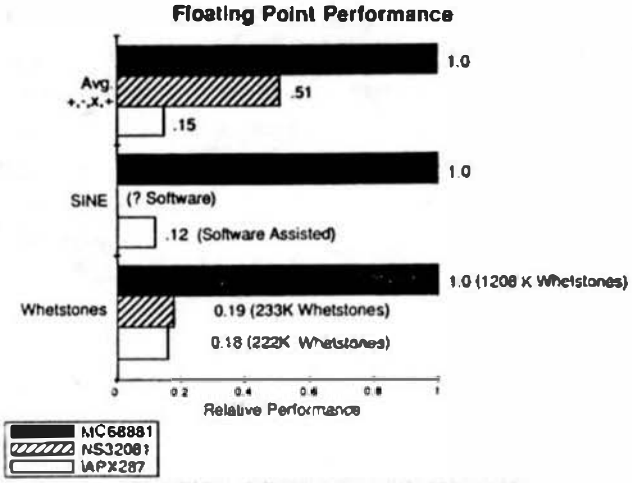
The MC68881 FPCP offers over 40 different floating point functions in hardware, more than any other coprocessor on the market (see attached figure). The MC68881 handles full extended precision (80 bits) for trigonometrics, hyperbolics, exponentials, logarithms, absolute values, square roots, etc., all in hardware. No software envelope is required (like other MPU families in the market). Also, the FPCP takes control of the floating point calculation completely, thus releasing the main processor (MC68020) to continue execution of other instructions. This feature is called concurrency which is a key technique used in supercomputers to increase system performance.

Floating point arithmetic is a daily requirement in everyday life since the real world cannot be defined by plain integer math. For example, the calculation of compound interest payments made to a savings account in a bank requires the use of floating point arithmetic. In many cases, the system's MPU will execute software which defines various floating point instructions. Normally these are the basic four functions (add, subtract, multiply, and divide), but more complex equations are required for both business and engineering applications. To improve the speed of these calculations the floating point functions are placed in hardware.

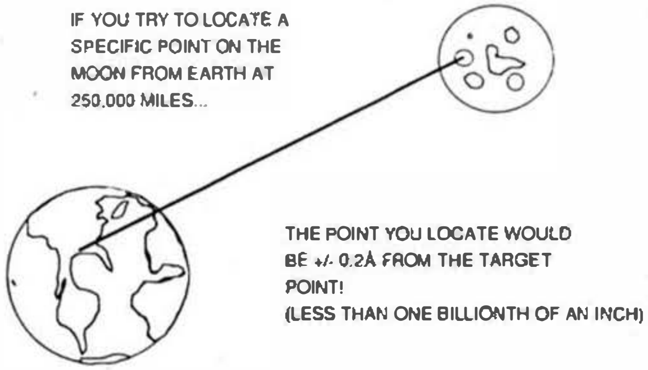
THE MC68881 FPCP IS NOT JUST ANOTHER 4-FUNCTION CALCULATOR

ARITHMETIC INSTRUCTIONS:	
FADD	Add
FSUB	Subtract
FINT	Take Integer part
FNEG	Negate
FNOP	No Operation (Sync)
FSQRT	Square Root
FGETMAN	Return Mantissa
FTST	Test the Operand
FGETEXP	Return Exponent
FCMP	Compare
FMUL	Multiply
FDIV	Divide
FMOD	Modulo
FREM	Remainder
FSCALE	Scale Exponent
FSGLMUL	Single Prec. Multiply
FSGLDIV	Single Prec. Divide
PLUS...	MOVES, BRANCHES, TRAPS, and MORE!

TRANSCENDENTAL INSTRUCTIONS	
FSIN	Sine
FASIN	Arc Sin
FSINH	Hyperbolic Sine
FCOS	Cosine
FACOS	Arc Cosine
FCOSH	Hyperbolic Cosine
FSINCOS	Simultaneous Sine/Cosine
FTAN	Tangent
FTANH	Hyperbolic Tangent
FATAN	Arc Tangent
FATANH	Hyperbolic Arc Tangent
FETOX	e to the XPower
FETOXM1	e to the (X-1) Power
FTENTOX	10 to the X Power
FTWOTOX	2 to the X Power
FLOG10	Logarithm base 10
FLOG2	Logarithm base 2
FLOGN	Logarithm base e
FLOGNP1	Logarithm base e of (X+1)



MC68881 CONSTANT ACCURACY


IF YOU TRY TO LOCATE A SPECIFIC POINT ON THE MOON FROM EARTH AT 250,000 MILES...

THE POINT YOU LOCATE WOULD BE +/- 0.2Å FROM THE TARGET POINT!
(LESS THAN ONE BILLIONTH OF AN INCH)

CLEARBROOK SOFTWARE GROUP
SOFTWARE UPDATE NOTICE

Version 1.3 of Clearbrook Software Group's Information Management System for OS9 6809 is now available. Its many enhancements include:

1. Faster by about 28%.
2. SORT statement to sort a data file.
3. SET SINGLE USER ON/OFF for faster operation on single user systems.
4. Reduced code size/increased data size.
5. Supplimentary manual (to replace READ.ME file).
6. Data/Index compatible with version 1.2 and CSG IMS 68000.

CSG IMS is now available for OS9 68000. Data and index files can be transported between 6809 version 1.3 and 68000 versions. Programs are source compatible between versions.

To get your free version 1.3 update, send your original version 1.2 disk (and your registration form if you have not returned it yet) to:

Clearbrook Software Group Inc.
Box 8000-499
Sumas, WA 98295
(604)853-9118

Gespac, inc

100 W. Hoover Ave. #11
Mesa, AZ 85202
(602) 962-5559
Telex 386 575

Dear Sir,

I am pleased to announce a new member to our fast growing family of G-64 bus products, a high performance local area controller board on a single height Eurocard.

This board will allow the system integrator a link several G-64 microcomputers together in a distributed processing architecture. This network board, in conjunction with others to be released soon, will provide G-64 bus users with a gateway to both VME bus and IBM-PC bus.

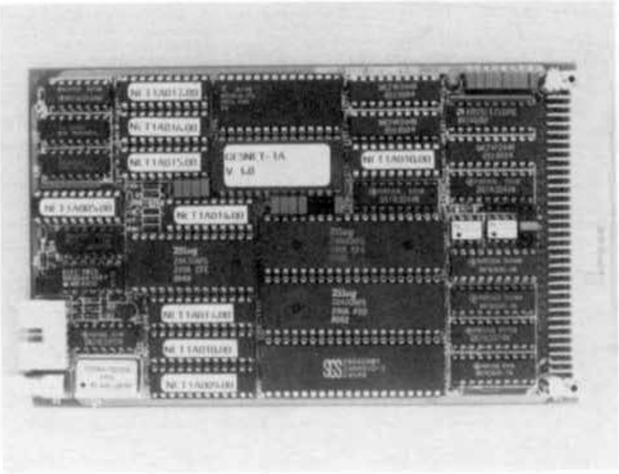
I would be grateful if you could inform the readers of your publication about this exciting new development. You will find enclosed a press kit we prepared to that purpose.

Please feel free to call me if you have any questions or need more information.

Sincerely yours,

Cosma Pabouctsidis
President

CP/t
Enclosure



GESPAC INTRODUCES INTELLIGENT LOCAL AREA NETWORK CONTROLLER CARD FOR THE G-64 BUS

Mesa, AZ, July 15, 1986--GESPAC Inc. introduces an intelligent local area network controller board, built on a single height Eurocard, and compatible with the standard G-64 bus.

The GESNET-1A allows the user to link up to 30 G-64 bus based microcomputer systems through a standard coaxial cable in a distributed processing environment. The board is ideally suited for industrial and process control applications.

Other versions of the GESNET-1A will soon be introduced by GESPAC for the IBM-PC bus and VME bus. This will allow the system integrator to link several front-end G-64 systems to a higher performance number cruncher based on VME bus, and/or to an inexpensive human interface based on the IBM-PC or clone.

The GESNET-1A uses a Base-Band NRZ data modulation mode at speeds up to 1 Megabits per second. The collision arbitration algorithm used by the GESNET-1A is a Collision Sense Multiple Access / Collision Avoidance (CSMA/CA) arbitration scheme. In this mode, when a collision is detected, the colliding modules will quickly compare their identification number in the data header. The lowest priority modules will remove themselves from the cable, allowing the highest priority module to take control of the network.

CSMA/CA is a very fast mechanism and will operate without difficulty on cable lengths of up to 1000 feet.

Above 1000 feet, the propagation delays induced by the cable affects the effectiveness of Collision Arbitration circuitry and the board will automatically switch to an Ethernet-Like CSMA/CD Collision Detection mode. The GESNET-1A can operate on cable up to 3000 feet long.

The GESNET-1A contains its own high speed Z80 microprocessor responsible for supervising the communications. The on-board firmware implements the first four layers of the seven layer Open System Interconnects specification. The GESNET-1A is capable of DMA and allows blocks of data to be transparently transferred from one system's memory to another's on the network.

The board will operate with any of the most performant 16-bit CPU available on the extended G-64 bus, such as the 68000, 68010 and 80286. The G-64 bus is a second generation 16-bit bus specifically aimed at midrange industrial application. Its compact single height Eurocard form factor and high reliability DIN-41612 pin-in-socket connector make it particularly resistant to hard environmental conditions.

The GESNET-1A is supported with software drivers which allow integration into MS-DOS when used with Intel microprocessors, and OS-9 when used with the 68000 family.

The board is available today at the low unit price of $995.

For more information contact

Joe Murphy
GESPAC Inc.
100 W. Hoover, Ave.
Mesa, AZ, 85202
(602) 962-5559

Dear Don,

In my previous letter I sort of left our readers hanging with two minor problems to consider, so before I carry on with XBASIC, maybe I should dispose of them first.

1. We were asked to convert the following :

```
   50     D=D+K%: IF I%<>J% THEN D=D*B%
```

into a single-statement line by using logic-functions instead of IF-THEN. Essentially, what we have to do is to multiply the new value of 'D' (as it exists at the point 'IF') by 1 if I%=J% and by B% if I%<>J%. This is accomplished by the following line :

```
   50     D=(D+K%) * (1-(B%-1)*(I%<>J%))
```

Here we see our original first statement enclosed in parens, and note that it is to be multiplied by '1'. This '1' is to be further modified 'IF I%<>J%)', which is how we read the part '*(I%<>J%)'. Earlier discussions will have taught how this will evaluate to '0' if 'I%<>J%)' is FALSE (ie I% is equal to J%), and to '-1' if it's TRUE. So, (B%-1) will be multiplied by '0' in the first instance, and by '-1' in the second, giving a net result of either (B%-1)*0, that is '0', or (B%-1)*-1, that is '-B%+1'. To complete the math then, our (D+K%) will be multiplied either by (1-0) or by 1-(-B%+1), which evaluates to 1 in the first case and to '1+B%-1, or 'B%' in the second. And hey presto! we've arrived. Hope this hasn't frightened you off, but it does enable us now to tack Line 50 on to the end of Line 40, and even Line 60 onto the end of that lot, if we so desire. All of which will save a little memory for us.

2. We had to figure out how to distinguish an erroneous '12' produced from VAL(12H45) from a genuine '12' entered in response to a request for numeric input. The solution is definitely not as awesome as that of our first example. Here it is :

```
  12  IF STR$(I) <> I$ GOTO 10
```

What we are doing here is to use our instant-camera 'STR$' to take a picture of 'I' (which may be a genuine '12' or a fake '12' produced from '12H45'). We then compare this picture with our original entry of I$, and if they don't match we know something has gone wrong, so we return for a new entry.

OK, enough of that! What shall we talk about next? Now about the different ways we can request input from the keyboard? Do you just use 'INPUT' as a matter of course, or do you sometimes consider the advantages (or disadvantages) of 'INPUT #0', or 'INPUT LINE' or even 'INCH$(0)'? Let's look at each in a little more detail ....

INPUT Fairly straightforward. It can be taken neat, as in 'INPUT I%' or 'INPUT I%,J$' for example, or it can print an input request message of some kind. Such as 'INPUT "Please enter your name and age",N$,A%'. In all cases it will display a '? ' (note the SPACE after the '?'), and pause for a response terminated by a CR, with commas to separate multiple responses. In the case of multiple requests, if the required number of responses is not made, further '? 's will be displayed until INPUT is satisfied. On the other hand, extra responses will be ignored! **INPUT will not accept a CR as a valid response.** The function of CR is to indicate 'End of reponses'.

Note, too, that a '?' on the end of "Please enter your name" is not appropriate, as we are not asking a question here, but issuing an instruction, whereas it would be OK on the end of "What is your name". So we come to

INPUT #0 which is almost the same as INPUT, except that it does not put up a trailing '?'. Its form is

```
50   INPUT #0, "Please enter your name ... ",N$
```

There are some differences, however, in that when you enter your name, followed by CR, the cursor does not move down to the following line, but simply returns to the left-hand margin. Should this input request then be followed by another, such as 'INPUT #0, "Your age ... ",A%', (after a response of, say, BOB to the first) the first message would be overlaid and you'd see (on the self-same line) the request :
'Your age ... your name ... BOB' with the cursor flashing over the 'y' of the second 'your'.

To correct this (unless you're using direct cursor-control to position your messages on the screen) you should follow your 'INPUT #0' request with a 'PRINT', thus 'INPUT #0, "Please enter your name ... ",N$: PRINT', which will cause the next message to be displayed on the succeeding line.

INPUT LINE The XBASIC manual tells us that this form is used to INPUT an entire line, including embedded SPACEs, etc. No messages can be displayed (as with INPUT and INPUT #0), and only one variable-name may be listed, so :

```
50  INPUT LINE A$  or  50  INPUT LINE B1$(5)
```

It displays a '? ', just as does INPUT, and normal testing of this form would indicate that there is no apparent difference between the two (apart from the restrictions mentioned), as INPUT will also accept a line of text as a response. There is one other difference though, which is not mentioned in the manual, and that is - INPUT LINE will accept a mere CR as a valid response, whereas the other two will not. They'll just keep coming back with further '? 's until satisfied. This could be useful in cases where you wish to set up a default response (defaults to CR) as in :

```
50  PRINT "Is your Printer a DOT-MATRIX* or THERMAL";:
    INPUT LINE R$
```

where the '*' indicates the default response. Observe that because INPUT LINE does not allow a message to be embedded, we cause it to be displayed by means of the PRINT statement, and then follow on with the INPUT LINE. Note the ';' immediately following the request message. This to nullify the CR which would normally occur on completion of the PRINT statement, so now the cursor is held at the end of the message, waiting for a response.

INCH$(0) This is another form which does not allow embedded messages. Unlike INPUT LINE, a single character response **without a following CR** is sufficient to meet its needs. Of course, the single response may itself be a mere CR! A very useful means of getting responses of the 'Y/N' type, or single-letter commands in a game, for instance. It is entered in the following pattern :

```
50  PRINT "Do you like this? (Y or N) ";: R$=INCH$(0)
60  IF R$="Y" GOTO xxxx ELSE GOTO yyyy
```

INCH$(0) does not put out any '?'s, and just like INPUT #0 it does not move to a different line after accepting input. So here again, unless you are using direct cursor-positioning, you would follow the INCH$(0) with a ': PRINT'. In closing, you should again note the ';' following the request-message. Its function is exactly the same as that described for INPUT LINE.

Next time maybe I'll talk a little about LSET and RSET amongst other things. I have never seen these used in normal XBASIC programs, other than in connection with FIELD statements where input is from a Data-File on disk.

Don Williams,
68 Micro Journal,
5900 Cassandra Smith Road,
Hixson, TN 37343

Sincerely,
Bob

R. Jones
President

PS How about starting a Rogue's Gallery of your more regular contributors? I've often wondered what Leo Taylor or Ron Anderson look like, for example. I'll start the ball rolling with a photo of myself taken a few years ago, in the days when I wore contact lenses. Unfortunately, the plastic had begun to crystallise after several years, and my corneas were starved for oxygen, as a result of which blood-vessels began to grow in to give them the necessary supply. Now I wear glasses, and the extra blood vessels have dried and become transparent.

PPS My re-write of all my XBASIC letters is coming along fine to date. The text has been expanded slightly, and I've gone into a little more explanation in some instances. In addition, I've covered other aspects of XBASIC, where it seemed appropriate, which have not been mentioned at all in my correspondence. When I get my first disk full, I'll send you 2 versions - - one in STYLO format, with emphasised print, etc., and the other in plain text form for those readers who don't have a word-processor.

Editor's Note: O.K. Bob, you are so right. I think a lot of the readers would like to know what you regular (and some not so regular) contributors look like.

We are preparing to run pictures, mostly scanned pictures from a Macintosh. So get those photos in folks!

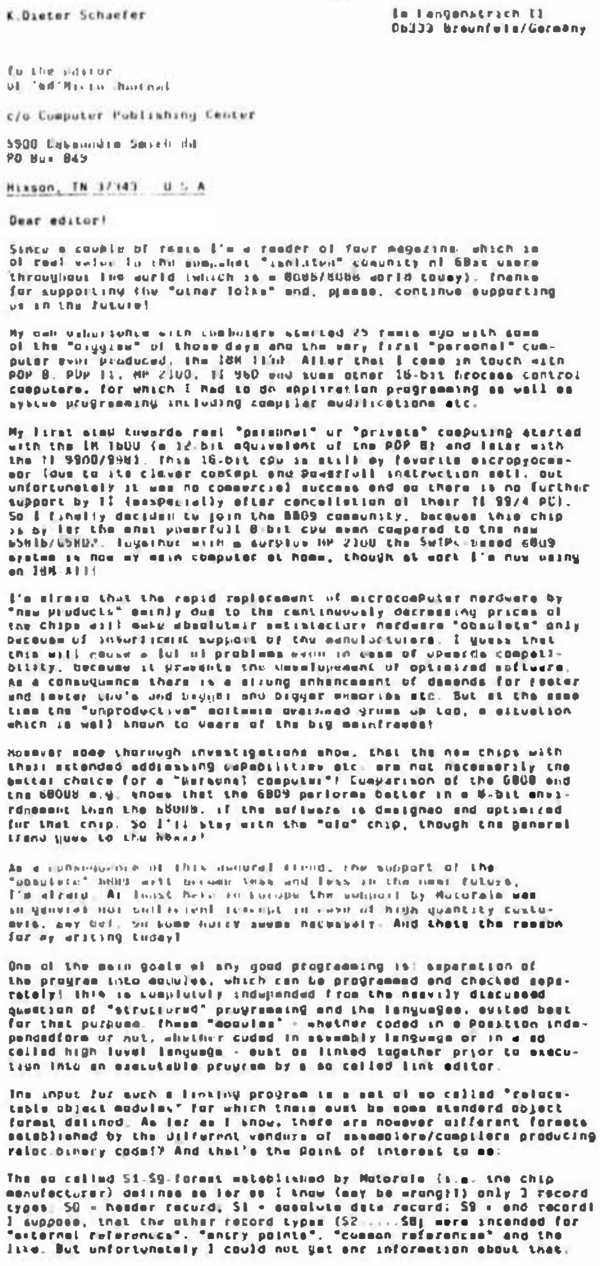
SOMETHING FOR ALL OF US / FROM ALL OF US

On the rewrite mentioned above; well, a lot of readers have remarked in letters and calls, that your BASIC offerings are appreciated. So keep 'em coming. The rewrite will allow a lot of those who missed some of the articles previously, a chance to catch up. Thanks Bob.

Also you readers who would like to have the rewrite, please drop a line and let me know, so I can figure how many to get printed up. The cost will be nominal, but no firm figure yet. We will have to wait and see just how big the entire package will be.

DMW

- - -



K.Dieter Schaefer

Im Langenstrach 11
6333 Braunfels/Germany

To the editor
of '68'Micro Journal

c/o Computer Publishing Center

5900 Cassandra Smith Rd
PO Box 849

Hixson, TN 37343 U.S.A

Dear editor!

Since a couple of years I'm a reader of your magazine, which is of real value to this somewhat "isolated" community of 68xx users throughout the world (which is a 8086/8088 world today). Thanks for supporting the "other folks" and, please, continue supporting us in the future!

My own experience with computers started 25 years ago with some of the "dinosaurs" of those days and the very first "personal" computer ever produced, the IBM 1130. After that I came in touch with PDP 8, PDP 11, HP 2100, TI 960 and some other 16-bit process control computers, for which I had to do application programming as well as system programming including compiler modifications etc.

My first step towards real "personal" or "private" computing started with the IM 1600 (a 12-bit equivalent of the PDP 8) and later with the TI 9900/9981. This 16-bit cpu is still my favorite microprocessor (due to its clever concept and powerfull instruction set), but unfortunately it was no commercial success and so there is no further support by TI (especially after cancellation of their TI 99/4 PC). So I finally decided to join the 6809 community, because this chip is by far the most powerfull 8-bit cpu even compared to the new 65816/65802. Together with a surplus HP 2100 the SWTPc based 6809 system is now my main computer at home, though at work I'm now using an IBM AT!!

I'm afraid that the rapid replacement of microcomputer hardware by "new products" mainly due to the continuously decreasing prices of the chips will make absolutely satisfactory hardware "obsolete" only because of insufficient support of the manufacturers. I guess that this will cause a lot of problems even in case of upwards compatibility, because it prevents the development of optimized software. As a consequence there is a strong enhancement of demands for faster and faster cpu's and bigger and bigger memories etc. But at the same time the "unproductive" software overhead grows up too, a situation which is well known to users of the big mainframes!

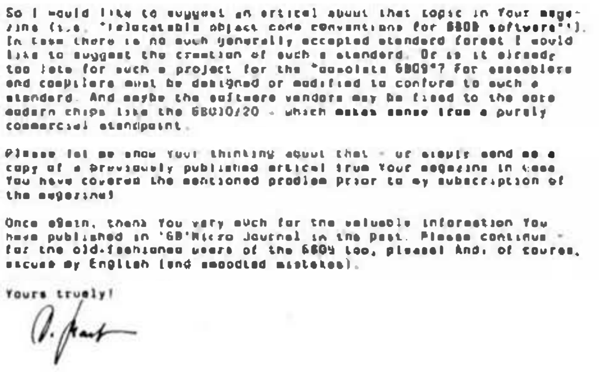
However some thorough investigations show, that the new chips with their extended addressing capabilities etc. are not necessarily the better choice for a "personal computer"! Comparison of the 6808 and the 68008 e.g. shows that the 6809 performs better in a 8-bit environment than the 68008, if the software is designed and optimized for that chip. So I'll stay with the "old" chip, though the general trend goes to the 68xxx!

As a consequence of this general trend, the support of the "obsolete" 6809 will become less and less in the near future, I'm afraid. At least here in Europe the support by Motorola was in general not sufficient (except in case of high quantity customers, any way), so some hurry seems necessary. And thats the reason for my writing today!

One of the main goals of any good programming is: separation of the program into modules, which can be programmed and checked separately! This is completely independend from the heavily discussed question of "structured" programming and the languages, suited best for that purpose. These "modules" - whether coded in a position independendform or not, whether coded in assembly language or in a so called high level language - must be linked together prior to execution into an executable program by a so called link editor.

The input for such a linking program is a set of so called "relocatable object modules" for which there must be some standard object format defined. As far as I know, there are however different formats established by the different vendors of assemblers/compilers producing reloc.binary code!? And that's the point of interest to me:

The so called S1-S9-format established by Motorola (i.e. the chip manufacturer) defines as far as I know (may be wrong!!) only 3 record types: S0 = header record, S1 = absolute data record; S9 = end record! I suppose, that the other record types (S2 ... S8) were intended for "external references", "entry points", "common references" and the like. But unfortunately I could not get any information about that.



So I would like to suggest an artical about that topic in your magazine (i.e. "relocatable object code conventions for 6809 software"!). In case there is no such generally accepted standard format I would like to suggest the creation of such a standard. Or is it already too late for such a project for the "obsolete 6809"? For assemblers and compilers must be designed or modified to conform to such a standard. And maybe the software vendors may be fixed to the more modern chips like the 68010/20 - which makes sense from a purely commercial standpoint.

Please let me know your thinking about that - or simply send me a copy of a previously published artical from your magazine in case you have covered the mentioned problem prior to my subscription of the magazine!

Once again, thank you very much for the valuable information you have published in '68'Micro Journal in the past. Please continue - for the old-fashioned users of the 6809 too, please! And, of course, excuse my English (and embedded mistakes).

Yours truely!

Editor's Note: Thanks for the letter above. We need your, and others input. I want to know what you want, in the way of future articles. Also I want some of you to let me know what you are willing and would like to write about. By getting both sides together, we all profit. And actually that is all I have been doing these past 9 years or so. Getting all of you together, authors and readers.

I was told, at NCC this year, by the Editor in Chief, of one of the larger computer magazines, that it was a 'miracle' we still existed! When I explained to him what it was (all of us) that made 68 MICRO JOURNAL and the way we go about it, he just shook his head and replied something to the effect that he did not realize that 'hackers like us' still existed.

Thinking about it later that night I realized that there is a big difference between the way each of us conducts his business. In his case it is mainly directed towards the advertisers (they actually foot the bill). His entire editorial and article (including reviews) policies, are so fabricated that few advertisers have cause to complain about their content. In our case we have stepped on some toes, losing some advertising as a result. But always after some of you were dealt with in a lesser manner that what we all expected, or were promised by their ads.

I bring this up only to shed some insight as to why we still support the 8 bitters. Because many of you still use them. And as long as most of you use them, we will be right in there as before. However, I do need some fresh input, in the way of articles. We need some along the lines of the letter above. We have a backlog of articles on hand that could carry us through a year or so, but do not address some of the subjects requested. And speaking of the 8 bitters, it seems that we need to share also with the newer 16 and 32 bitters. However, it all needs to be kept in perspective. Most times you let me know if I allow that to slip too far.

So, please let me know what YOU can contribute to OUR magazine. It certainly will help Larry and I plan for the coming months.

If I had attempted to run 68 MICRO JOURNAL, as my 'Editor in Chief friend' does his magazine, we would have been gone years ago! By ourselves, without you, 68 MICRO JOURNAL would not exist. You are 68 MICRO JOURNAL!

How 'bout it?

DMW

- - -

CONTINUED FROM LAST MONTH

The Editor,
Computer Publishing Center,
68 Micro Journal,
5900 Cassandra Smith,
P.O. Box 849,
Hixson, TN 37343,
U.S.A.

Chemistry Department
University of Transkei
Private Bag X5092
UMTATA
Republic of Transkei
Southern Africa

```
                 * DATIR.CMD -  File protection.
                 * by LPL Piacenza, (c) April 1985.
                 * Chemistry Dept., University of Transkei.
                 *  for PRIVATE/PERSONAL use only.
                 *
                 * Will WRITE, DELETE or CATALOGUE protect
                 * the entire disk or selected files.
                 * Will also de-protect entire disk or selected files.
                 * Existing protection status: OPTIONALLY retained.
                 *
                 * Can USE '?' as WILD CARD character
                 * for file names: e.g. EX???T.D?R  or ??AM.9?N
                 *
                 * OPTIONS   W,D,C,X   for
                 * Write, Delete, Catalogue, Remove all protection.
                 *
                 * SYNTAX DATIR,<drive>,Filespecs,+codes (W,D,C,X)
                 *
                 * for WHOLE DISK, USE: DATIR,<drive>,?,+codes
                 *  or (faster)         DATIR,<drive>,+codes
                 *
                 * PROMPT: Y/N/A/R/Q = yes/no/automatic/retain/quit
                 *
                 * If EXISTING protection is to be RETAINED for a
                 * file regardless of new options (except X which
                 * overrides all options anyway),
                 *  then use 'R' instead of 'Y' at the prompt.
                 *
                 * In auto mode use ESC key to exit to Flex.


           CB40  SYSFCB  EQU    $CB40
           CB43  SYSDRV  EQU    SYSFCB+3
           CB44  SYSNAM  EQU    SYSFCB+4
           CB4B  SYSEXT  EQU    SYSFCB+11
           CC02  TTYEOL  EQU    $CC02
           CC0C  WDRIVE  EQU    $CC0C
           CC11  LSTTRM  EQU    $CC11
           CC14  BUFPNT  EQU    $CC14
           CD03  WARMS   EQU    $CD03
           CD15  GETCHR  EQU    $CD15
           CD18  PUTCHR  EQU    $CD18
           CD1B  INBUFF  EQU    $CD1B
           CD1E  PSTRNG  EQU    $CD1E
           CD24  PCRLF   EQU    $CD24
           CD27  NXTCHR  EQU    $CD27
           CD3F  RPTERR  EQU    $CD3F
           CD42  GETHEX  EQU    $CD42
           CD4E  STAT    EQU    $CD4E
           D3E5  INCHNE  EQU    $D3E5        input without echo.
           D406  FMS     EQU    $D406

                 * Miscellaneous equates.

           0008  EOFIL   EQU    8
           000D  CR      EQU    13
           003F  WILD    EQU    '?           wild card character.
           001B  ESC     EQU    27           exit to Flex from auto-mode.
           002D  NOPROT  EQU    '-
           0059  YES     EQU    'Y
           0041  AUT     EQU    'A           response for automatic.
           0052  KEEP    EQU    'R           retain old protection.
           0051  STOP    EQU    'Q           quit response.
           005F  UPPER   EQU    $5F          force upper case.
           0020  SPACE   EQU    32
           002C  COMMA   EQU    ',
           002E  POINT   EQU    '.

                 * Setup offsets from U register.

0000                     ORG    0

0000             PROT    RMB    1            protection bits.
0001             AUTO    RMB    1            auto-process flag.
0002             FDONE   RMB    1            all done flag.
0003             PLUSGN  RMB    2            adrs of '+'(+1) in linbuff.
0005             BUFP1   RMB    2            temp. bufpointer address.
0007             BUFP2   RMB    2            temp. bufpointer address.
0009             PFLAG   RMB    1            flag that prot. codes found.
000A             FOPEN   RMB    1            flag that DIR is open.
000B             NAM     RMB    8            filename bytes.
0013             EXT     RMB    3            extension bytes.
0016             PWRITE  RMB    1            current write-prot. status.
0017             PDEL    RMB    1            current delete-prot. status.
0018             PCAT    RMB    1            current cat-prot. status.
0019             EXIST   RMB    1            flags keep previous status.
001A             DOALL   RMB    1            do all files.
           001A  OFFSET  EQU    *-1          max. U offset from S(stack).

C100                     ORG    $C100

C100 20    19    DATIR   BRA    START

C102 83 2E 85            FCB    $83,$2E,$85
C105 3A 20 28 63         FCC    ': (c)1985, L. Piacenza'
C109 29 31 39 38
C10D 35 2C 20 4C
C111 2E 20 50 69
C115 61 63 65 6E
C119 7A 61

                 * Setup U pointer and clear that memory.

C11B             START
C11B 32    E8 E6         LEAS   -OFFSET,S
C11E 33    E4            LEAU   ,S
C120 C6    1A            LDB    #OFFSET
C122 4F                  CLRA
C123 1F    8B            TFR    A,DP
C125             CLEARU
C125 A7    C5            STA    B,U
C127 5A                  DECB
C128 2A    FB            BPL    CLEARU   include zero offset!

C12A 17    01FA          LBSR   STORE    store buffer pointer.

                 * GET DRIVE NUMBER IF SPECIFIED IN THE COMMAND LINE.
                 * If illegal or no drive specified, then default to
                 * working drive.

C12D 17    019C          LBSR   TTYLST   check last terminator.
C130 1027 026C           LBEQ   HELP     no parameters!
C134 81    20            CMPA   #' 
```

```
 C136 26   03                BNE   GETDRV
 C138 63   C8 1A             COM   DOALL,U    do all files.
 C13B                GETDRV
 C13B BD   CD42              JSR   GETHEX
 C13E 25   09                BCS   GETDR      got a number/letter?
 C140 5D                     TSTB
 C141 27   06                BEQ   GETDR
 C143 1F   10                TFR   X,D
 C145 C1   04                CMPB  #4         valid drive?
 C147 25   08                BCS   DRVNUM     yes.
 C149                GETDR
 C149 AE   45                LDX   BUFP1,U    reset buffer pointer.
 C14B BF   CC14              STX   BUFPNT
 C14E F6   CC0C      DFLTD   LDB   WDRIVE     use working drive.
 C151 8E   C840      DRVNUM  LDX   #SYSFCB
 C154 E7   03                STB   3,X        drive number set in FCB.

 C156 17   01CE              LBSR  STORE      save current buf-pointer.
 C159 17   0178              LBSR  PROTECT    get protection codes
 C15C 6D   49                TST   PFLAG,U    if 0, no options found!
 C15E 26   06                BNE   FSPECS     codes ARE in command line.

 C160 BD   CD24      TOFLEX  JSR   >PCRLF
 C163 7E   CD03              JMP   >WARMS

 C166                FSPECS
 C166 6D   C8 1A             TST   DOALL,U    do all files?
>C169 1020 0069              LBMI  PROCES     yes, skip comparing.

 C16D                SCAN
 C16D 6D   42                TST   FDONE,U    all entries done?
 C16F 26   EF                BNE   TOFLEX     yes, exit.
 C171 17   014D              LBSR  CLEAR      clear filename.ext bytes.
 C174 5F                     CLRB             name byte count: max=8.
 C175 31   4B                LEAY  NAM,U
 C177 AE   45                LDX   BUFP1,U
 C179                SCAN2
 C179 8D   1C                BSR   CHARIN
 C17B 81   2E                CMPA  #POINT     is it extension?
 C17D 27   09                BEQ   SCEXT
 C17F 5C                     INCB             count valid characters.
 C180 C1   09                CMPB  #9
 C182 24   29                BCC   NAMERR     too many!
 C184 A7   A0                STA   ,Y+        store them.
 C186 20   F1                BRA   SCAN2

 C188                SCEXT
 C188 31   C8 13             LEAY  EXT,U      do extension specs.
 C18B 5F                     CLRB             max=3 chars.
 C18C                SCEXT2
 C18C 8D   09                BSR   CHARIN
 C18E 5C                     INCB
 C18F C1   04                CMPB  #4
 C191 24   1A                BCC   NAMERR     too many!
 C193 A7   A0                STA   ,Y+
 C195 20   F5                BRA   SCEXT2

 C197                CHARIN
 C197 8D   08                BSR   FETCH
 C199 27   33                BEQ   EOLS       command line done?
 C19B 81   20                CMPA  #SPACE
 C19D 27   11                BEQ   NAMDON     separator found
 C19F 81   2C                CMPA  #COMMA
 C1A1 27   0D                BEQ   NAMDON
 C1A3 39                     RTS

 C1A4                FETCH
 C1A4 BD   CD27              JSR   NXTCHR
 C1A7 BE   CC14              LDX   BUFPNT     keep tabs on position.
 C1AA 16   0122              LBRA  FIXLIN     indirect return.

 C1AD                NAMERR
 C1AD 16   0104              LBRA  SYNTAX     report 'syntax' error.

 C1B0                NAMDON
 C1B0 32   62                LEAS  2,S
 C1B2 8E   CC14              LDX   BUFPNT     save buffer pointer
 C1B5 AF   47                STX   BUFP2,U    for next scan.
 C1B7 20   1D                BRA   PROCES

 C1B9                EOF
 C1B9 6D   C8 1A             TST   DOALL,U    if eof when doing all
>C1BC 102B FFA0              LBMI  TOFLEX     files, then finished!
 C1C0 AE   47                LDX   BUFP2,U    on end of file
 C1C2 AF   45                STX   BUFP1,U    reset pointers,
 C1C4 BF   CC14              STX   BUFPNT
 C1C7 6F   4A                CLR   FOPEN,U    re-open DIR,
 C1C9 BD   CD24              JSR   PCRLF      print a blank line,
 C1CC 20   9F                BRA   SCAN       & repeat process.

 C1CE                EOLS
 C1CE 32   62                LEAS  2,S
 C1D0 AC   45                CMPX  BUFP1,U    if at the same pos,
 C1D2 27   8C                BEQ   TOFLEX     then all done!
 C1D4 6C   42                INC   FDONE,U    tell current specs done.

 C1D6                PROCES
 C1D6 6D   4A                TST   FOPEN,U    if DIR open,
 C1D8 26   0B                BNE   FILNAM     then skip this part.
 C1DA 86   06                LDA   #6         open DIR.
 C1DC 17   009E              LBSR  FMSCAL
 C1DF 1026 00D8              LBNE  ERRORS
 C1E3 6C   4A                INC   FOPEN,U    tell DIR open.
 C1E5                FILNAM
 C1E5 86   07                LDA   #7         get a directory entry.
 C1E7 17   0093              LBSR  FMSCAL
 C1EA 1026 00BA              LBNE  ERRORD     eof?
 C1EE 6D   04                TST   4,X
 C1F0 2B   F3                BMI   FILNAM     skip deleted files.
 C1F2 27   C5                BEQ   EOF        unused, must be end of dir.
 C1F4 6D   C8 1A             TST   DOALL,U
 C1F7 2B   1A                BMI   FOUND

                     * now match this file with specs.

 C1F9 31   4B                LEAY  NAM,U
 C1FB 8E   C843              LDX   #SYSDRV
 C1FE C6   0C                LDB   #12
 C200                MATCH
 C200 5A                     DECB             match 11 bytes.
 C201 27   10                BEQ   FOUND
 C203 30   01                LEAX  1,X
 C205 A6   A0                LDA   ,Y+
 C207 27   F7                BEQ   MATCH
 C209 81   3F                CMPA  #WILD      wild card char?
 C20B 27   F3                BEQ   MATCH
 C20D A1   84                CMPA  ,X
 C20F 27   EF                BEQ   MATCH
 C211 20   D2                BRA   FILNAM     no match, get another entry.

                     * a match found.

 C213                FOUND
 C213 8D   78                BSR   QUIT       check for ESC key.
 C215 30   8D 0140           LEAX  CURRNT,PCR tell current protection
 C219 BD   CD1E              JSR   PSTRNG     codes for this file.
 C21C 17   0115              LBSR  TELPRT
 C21F 30   8D 014E           LEAX  ATRMSG,PCR ask for change.
 C223 8D   73                BSR   OUTP4
 C225 8E   C844              LDX   #SYSNAM    write filename,
 C228 C6   08                LDB   #8
 C22A 8D   73                BSR   BOUT
 C22C 86   2E                LDA   #POINT
 C22E BD   CD18              JSR   PUTCHR
 C231 C6   03                LDB   #3         and extension.
 C233 8D   6A                BSR   BOUT
```

```
C235 86   59                 LDA    #YES
C237 6D   41                 TST    AUTO,U    auto mode active?
C239 26   29                 BNE    AUTOM     yes.
C23B 30   8D 0151            LEAX   QUERY,PCR prompt.
C23F 8D   57                 BSR    OUTP4
C241 BD   CD15               JSR    GETCHR
C244 84   5F                 ANDA   #UPPER
C246 81   51                 CMPA   #STOP     quit?
C248 1027 FF14               LBEQ   TOFLEX
C24C 81   41                 CMPA   #AUT      automatic henceforth?
C24E 26   04                 BNE    MANUAL
C250 A7   41                 STA    AUTO,U    yes, automatic.
C252 20   10                 BRA    AUTOM

C254                MANUAL
C254 6F   C8 19              CLR    EXIST,U   unique response unless auto.
C257 81   52                 CMPA   #KEEP     retain existing codes?
C259 26   05                 BNE    FND2
C25B                RETAIN
C25B A7   C8 19              STA    EXIST,U   set flag for retain.

C25E 20   04                 BRA    AUTOM

C260                FND2
C260 81   59                 CMPA   #YES      yes?
C262 26   01                 BNE    FILNAM
C264                AUTOM
C264 8E   C840               LDX    #SYSFCB
C267 A6   C4                 LDA    PROT,U    change protection bits
C269 27   07                 BEQ    NOVLAY    'X' overrides everything!
C26B 6D   C8 19              TST    EXIST,U
C26E 27   02                 BEQ    NOVLAY    don't keep existing bits.
C270 AA   0F                 ORA    15,X      add to existing bits.
C272                NOVLAY
C272 A7   0F                 STA    15,X      for this dir. entry,
C274 86   0B                 LDA    #B        & rewrite the entry,
C276 8D   05                 BSR    FMSCAL    so that DIR. is
C278 26   41                 BNE    ERRORS    updated.
C27A 16   FF68               LBRA   FILNAM

C27D                FMSCAL
C27D 8E   C840               LDX    #SYSFCB
C280 A7   84                 STA    ,X
C282 7E   D406               JMP    FMS       indirect return.

C285                QUIT
C285 BD   CD4E               JSR    STAT
C288 27   0A                 BEQ    QUIRET
C28A AD   9F D3E5            JSR    [INCHNE]  input with no echo.
C28E 81   1B                 CMPA   #ESC
C290 1027 FECC               LBEQ   TOFLEX

C294                QUIRET
C294 39                      RTS

                    * output a string terminated with $04.
                    * enter with IX pointing to string.

C295 BD   CD18      OUTP     JSR    >PUTCHR
C298 A6   80        OUTP4    LDA    ,X+
C29A 81   04                 CMPA   #4
C29C 26   F7                 BNE    OUTP
C29E 39                      RTS

                    * output a string of ACCB bytes.
                    * enter with ACCB=byte count &
                    * with IX pointing to string.


C29F                BOUT
C29F A6   80                 LDA    ,X+
C2A1 BD   CD18               JSR    PUTCHR
C2A4 5A                      DECB
C2A5 26   F8                 BNE    BOUT
C2A7 39                      RTS

                    * test for end-of-file.

C2A8                ERRORB
C2A8 8E   C840               LDX    #SYSFCB
C2AB A6   01                 LDA    1,X
C2AD 81   08                 CMPA   #EOFIL
C2AF 26   0A                 BNE    ERRORS
C2B1 16   FF05               LBRA   EOF

C2B4                SYNTAX
C2B4 86   1A                 LDA    #26
C2B6 8E   C840               LDX    #SYSFCB
C2B9 A7   01                 STA    1,X
C2BB BD   CD3F      ERRORS   JSR    >RPTERR   report all fatal errors,
C2BE 7E   CD03               JMP    >WARMS    & return to FLEX.

                    * clear 11 name+extension bytes in U offset.

C2C1                CLEAR
C2C1 30   48                 LEAX   NAM,U
C2C3 C6   0B                 LDB    #11
C2C5 4F                      CLRA
C2C6                CLOOP
C2C6 A7   80                 STA    ,X+
C2C8 5A                      DECB
C2C9 26   FB                 BNE    CLOOP
C2CB 39                      RTS

                    * Check for end of line in input buffer.
                    * returns with ZERO condition (BEQ) if end of line,
                    * else BNE, not equal.
                    * TTYEOL is FLEX end of line, usually ':'

C2CC                TTYLSI
C2CC B6   CC11               LDA    LSTTRM    entry point.
C2CF                TTYLIN
C2CF 81   0D                 CMPA   #CR       alternate entry point.
C2D1 27   03                 BEQ    TTYRET
C2D3 B1   CC02               CMPA   TTYEOL
C2D6 39             TTYRET   RTS

                    * Find protection specifications in command line.

C2D7                PROTECT
C2D7 AE   47                 LDX    BUFP2,U   reset buffer pointer.
C2D9 BF   CC14               STX    BUFPNT
C2DC                PLUS
C2DC BD   CD27               JSR    NXTCHR
C2DF 8D   EE                 BSR    TTYLIN
C2E1 27   3E                 BEQ    RETURN
C2E3 81   2B                 CMPA   #'+
C2E5 26   F5                 BNE    PLUS
C2E7 BE   CC14               LDX    BUFPNT
C2EA AF   43                 STX    PLUSGN,U  address of 1st prot. char.
C2EC                GETSPEC
C2EC BD   CD27               JSR    NXTCHR
C2EF 8D   DE                 BSR    TTYLIN
C2F1 27   28                 BEQ    RET1
C2F3 E6   C4                 LDB    PROT,U
C2F5 84   5F                 ANDA   #UPPER
```

```
C2F7 81      44                  CMPA   #'D       delete protect?
C2F9 27      10                  BEQ    SETD
C2FB 81      43                  CMPA   #'C       catalogue protect?
C2FD 27      10                  BEQ    SETC
C2FF 81      58                  CMPA   #'X       de-protect?
C301 27      14                  BEQ    SETX
C303 81      57                  CMPA   #'W       write protect?
C305 26      E5                  BNE    GETSPEC

C307                     SETW
C307 CA      80                  ORB    #$80
C309 20      06                  BRA    SETP

C30B                     SETD
C30B CA      40                  ORB    #$40
C30D 20      02                  BRA    SETP

C30F                     SETC
C30F CA      10                  ORB    #$10
C311                     SETP
C311 E7      C4                  STB    PROT,U
C313 E7      49                  STB    PFLAG,U   tell prot. codes found.
C315 20      05                  BRA    GETSPEC

C317                     SETX
C317 6F      C4                  CLR    PROT,U    clear prot. bits.
C319 A7      49                  STA    PFLAG,U   tell prot. codes found.

C31B                     REST
C31B AE      43                  LDX    PLUSGN,U
C31D 86      0D                  LDA    #CR       force command line end
C31F A7      82                  STA    ,-X       where '+' was.
C321                     RETURN
C321 AE      47                  LDX    BUFP2,U
C323 BF      CC14                STX    BUFPNT    rescan from 1st file spec.
C326 39                          RTS

C327                     STORE
C327 BE      CC14                LDX    BUFPNT    make duplicates of
C32A AF      45                  STX    BUFP1,U   buffer pointer.
C32C AF      47                  STX    BUFP2,U
C32E A6      84                  LDA    ,X
C330 B7      CC11                STA    LSTTRM
C333 39                          RTS

                         * Print current protection codes for this file.
                         * A '-' means NO protection currently in effect.

C334                     TELPRT
C334 86      20                  LDA    #NOPROT
C336 1F      89                  TFR    A,B
C338 ED      C8 16               STD    PWRITE,U
C33B E7      C8 18               STB    PCAT,U
C33E BE      C840                LDX    #SYSFCB
C341 E6      0F                  LDB    15,X
C343 86      57                  LDA    #'W
C345 C5      80                  BITB   #$80      is file write-protected?
C347 27      03                  BEQ    TEL2
C349 A7      C8 16               STA    PWRITE,U
C34C                     TEL2
C34C 86      44                  LDA    #'D       is file delete-protected?
C34E C5      40                  BITB   #$40
C350 27      03                  BEQ    TEL3
C352 A7      C8 17               STA    PDEL,U
C355                     TEL3
C355 86      43                  LDA    #'C       is file cat. protected?
C357 C5      10                  BITB   #$10
C359 27      03                  BEQ    TELRET
C35B A7      C8 18               STA    PCAT,U
C35E                     TELRET
C35E 30      C8 16               LEAX   PWRITE,U  print current
C361 C6      03                  LDB    #3        protection.
C363 16      FF39                LBRA   BOUT

C366 43 75 72 72         CURRNT  FCC    'Currently ',4
C36A 65 6E 74 6C
C36E 79 20 04
C371 20 3A 20 43         ATRMSG  FCC    ' : Change Attributes of '
C375 68 61 6E 67
C379 65 20 41 74
C37D 74 72 69 62
C381 75 74 65 73
C385 20 6F 66 20
C389 66 69 6C 65                 FCC    'file '
C38D 20
C38E 22 04                       FCB    $22,4
C390 22                  QUERY   FCB    $22
C391 20 28 59 2F                 FCC    ' (Y/N/R/A/Q)? ',4
C395 4E 2F 52 2F
C399 41 2F 51 29
C39D 3F 20 04

C3A0                     HELP
C3A0 30      8D 0006             LEAX   HLP1,PCR
C3A4 BD      CD1E                JSR    PSTRNG
C3A7 7E      CD03                JMP    WARMS

C3AA 0C 55 73 65         HLP1    FCC    12,'Use PLUS sign (+)'
C3AE 20 50 4C 55
C3B2 53 20 73 69
C3B6 67 6E 20 28
C3BA 2B 29
C3BC 20 69 6E 20                 FCC    ' in front of protection'
C3C0 66 72 6F 6E
C3C4 74 20 6F 66
C3C8 20 70 72 6F
C3CC 74 65 63 74
C3D0 69 6F 6E
C3D3 20 6F 70 74                 FCC    ' options,'
C3D7 69 6F 6E 73
C3DB 2C
C3DC 0A 0D                       FCB    10,13
C3DE 20 65 67 3A                 FCC    ' eg: DATTR,<drive>,FILES.TXT,+W'
C3E2 20 44 41 54
C3E6 54 52 2C 3C

C3EA 64 72 69 76
C3EE 65 3E 2C 46
C3F2 49 4C 45 53
C3F6 2E 54 58 54
C3FA 2C 2B 57
C3FD 0A 0A 0D 55                 FCC    10,10,13,'Use ? as wild card'
C401 73 65 20 3F
C405 20 61 73 20
C409 77 69 6C 64
C40D 20 63 61 72
C411 64
C412 2C 20 65 67                 FCC    ', eg: DATTR,F???S.T?T,+D'
C416 3A 20 44 41
C41A 54 54 52 2C
C41E 46 3F 3F 3F
C422 53 2E 54 3F
C426 54 2C 2B 44
C42A 0A 0A 0D 55                 FCC    10,10,13,'Use ? or no specs'
C42E 73 65 20 3F
C432 20 6F 72 20
C436 6E 6F 20 73
C43A 70 65 63 73
```

```
493   C43E 20 74 6F 20          FCC    ' to process entire'
      C442 70 72 6F 63
      C446 65 73 73 20
      C44A 65 6E 74 69
      C44E 72 65
494   C450 20 64 69 73          FCC    ' disk, eg: DATTR,?,+WC'
      C454 6B 2C 20 65
      C458 67 3A 20 44
      C45C 41 54 54 52
      C460 2C 3F 2C 20
      C464 57 43
495   C466 20 20 20 20          FCC    '    or DATTR,+WC'
      C46A 6F 72 20 44
      C46E 41 54 54 52
      C472 2C 20 57 43
496   C476 0A                   FCB    10
497   C477 0A 0A 0D 57          FCC    10,10,13,'W=write protect, '
      C47B 3D 77 72 69
      C47F 74 65 20 70
      C483 72 6F 74 65
      C487 63 74 2C 20
498   C48B 44 3D 64 65          FCC    'D=delete protect, '
      C48F 6C 65 74 65
      C493 20 70 72 6F
      C497 74 65 63 74
      C49B 2C 20
499   C49D 43 3D 63 61          FCC    'C=catalogue protect.'
      C4A1 74 61 6C 6F
      C4A5 67 75 65 20
      C4A9 70 72 6F 74
      C4AD 65 63 74 2E
500   C4B1 0A 0D 58 3D          FCC    10,13,'X=remove all protection.'
      C4B5 72 65 6D 6F
      C4B9 76 65 20 61
      C4BD 6C 6C 20 70
      C4C1 72 6F 74 65
      C4C5 63 74 69 6F
      C4C9 6E 2E
501   C4CB 0A                   FCB    10
502   C4CC 0A 0D 52 65          FCC    10,13,'Response Y(es),'
      C4D0 73 70 6F 6E
      C4D4 73 65 20 59
      C4D8 28 65 73 29
      C4DC 2C
503   C4DD 4E 28 6F 29          FCC    'N(o),R(etain),A(utomatic),'
      C4E1 2C 52 28 65
      C4E5 74 61 69 6E
      C4E9 29 2C 41 28
      C4ED 75 74 6F 6D
      C4F1 61 74 69 63
      C4F5 29 2C
504   C4F7 51 28 75 69          FCC    'Q(uit)'
      C4FB 74 29
505   C4FD 0A 0D 52 3A          FCC    10,13,'R: merges existing status'
      C501 20 6D 65 72
      C505 67 65 73 20
      C509 65 78 69 73
      C50D 74 69 6E 67
      C511 20 73 74 61
      C515 74 75 73
506   C518 20 77 69 74          FCC    ' with new protection.'
      C51C 68 20 6E 65
      C520 77 20 70 72
      C524 6F 74 65 63
      C528 74 69 6F 6E
      C52C 2E
507   C52D 0A 0D 20 52          FCC    10,13,' R mode remains active if'
      C531 20 6D 6F 64
      C535 65 20 72 65
      C539 6D 61 69 6E
      C53D 73 20 61 63
      C541 74 69 76 65
      C545 20 69 66
508   C548 20 75 73 65          FCC    ' used prior to Auto mode.'
      C54C 64 20 70 72
      C550 69 6F 72 20
      C554 74 6F 20 41
      C558 75 74 6F 20
      C55C 6D 6F 64 65
      C560 2E
509   C561 0A 0D 41 3A          FCC    10,13,'A: processes all matched'
      C565 20 70 72 6F
      C569 63 65 73 73
      C56D 65 73 20 61
      C571 6C 6C 20 6D
      C575 61 74 63 68
      C579 65 64
510   C57B 20 66 69 6C          FCC    ' files automatically. '
      C57F 65 73 20 61
      C583 75 74 6F 6D
      C587 61 74 69 63
      C58B 61 6C 6C 79
      C58F 2E 20
511   C591 55 73 65 20          FCC    'Use Escape key to quit A mode.'
      C595 45 73 63 61
      C599 70 65 20 6B
      C59D 65 79 20 74
      C5A1 6F 20 71 75
      C5A5 69 74 20 41
      C5A9 20 6D 6F 64
      C5AD 65 2E
512   C5AF 04                   FCB    4
513
514                             END    DATTR

0 ERROR(S) DETECTED

SYMBOL TABLE:

ATRMSG C371   AUT    0041   AUTO   0001   AUTOM  C264   BOUT   C29F
BUFP1  0005   BUFP2  0007   BUFPNT CC14   CHARIN C197   CLEAR  C2C1
CLEARU C125   CLOOP  C2C6   COMMA  002C   CR     000D   CURRNT C366
DATTR  C100   DFLTO  C14E   DOALL  001A   DRVNUM C151   EOF    C109
EOFIL  0008   EOLS   C1CE   ERRORB C2A0   ERRORS C20B   ESC    001B
EXIST  0019   EXIT   0013   FBONE  0002   FETCH  C1A4   FILNAM C1E5
FMS    D406   FMSCAL C27B   FNB2   C260   FOPEN  000A   FOUND  C213
FSPECS C166   GETCHR CD15   GETDR  C149   GETDRV C138   GETHEX CD42
GETSPE C2EC   HELP   C3A0   HELP1  C3AA   INBUFF CD10   INCHNE D3E5
KEEP   0052   LSTTRM CC11   MANUAL C254   MATCH  C200   MAM    0000
MARBON C1B0   NAMERR C1AD   NOPROT 002D   NOVLAY C272   NXTCHR CD27
OFFSET 001A   OUTP   C295   OUTP4  C29B   PCAT   0018   PCRLF  CD24
PDEL   0017   PFLAG  0009   PLUS   C2DC   PLUSGN 0003   POINT  002E
PROCES C1D6   PROT   0000   PROTEC C287   PSTRNG CD1E   PUTCHR CD18
PWRITE 0016   QUERY  C390   QUIRET C294   QUIT   C205   RETI   C31B
RETAIN C258   RETURN C321   RPTERR CD3F   SCAN   C168   SCAN2  C179
SCEXIT C188   SCEXT2 C18C   SETC   C30F   SETD   C300   SETP   C311
SETW   C307   SETX   C317   SPACE  0020   START  C11D   STAT   CD4E
STOP   0051   STORE  C327   SYNTAX C204   SYSDRV C843   SYSEXT C848
SYSFCB C84B   SYSNAM C844   TEL2   C34C   TEL3   C355   TELPRT C334
TELRET C35E   TOFLEX C160   TTYEOL CC02   TTYLIN C2CF   TTYLST C2CC
TTYRET C2D6   UPPER  005F   WARMS  CD03   WDRIVE CC0C   WILD   003F
YES    0059
```

Classifieds *As submitted - No Guarantees*

Winchester 10 Megabyte Drives
Two (2) 10 Megabyte Hard-Disk Winchester Drives. Working - were removed for upgrade to larger drives.
1 - Seagate Model #412 $275.00
3 - Dual 8" drive enclosure with power supply. New in box. $175.00 each.
5 - Siemens 8" Disk Drives. $150.00 each.
Tano Outpost II, 56K, 2 5" DSDD Drives, FLEX, MUMPS $595.
MICROKEY Single Board Computer, Target 128K RAM, FLEX, FORTH, with optional 6502 CPU & ROMS as advertised on p. 51 DEC. 84 68' Micro Journal. $1800.
TELETYPE Model 43 PRINTER - with serial (RS232) interface and full ASCII keyboard. $359.00 ready to run.
SWTPC S/09 with Motorola 128K RAM, 1-MPS2, 1-Parallel Port, MP-09 CU Card $1290.
1-CDSI 20 Meg Hard Disk System with controller $1000.
(615) 842-4600 M-F 9 AM to 5 PM EST

•••

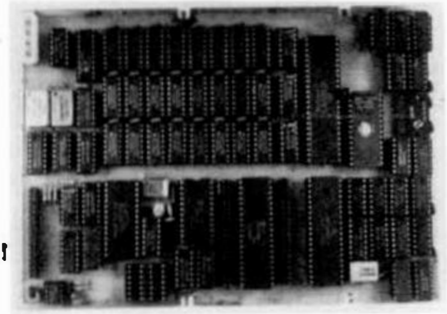
FOR SALE: SWTPC S+ w/256K; A1 streaming tape; DMF3; F-8" floppy disk drive; CDS-3 80mb drive w/controller; 3 Microterm terminals; DEC LA100ZA printer; quad serial board; SWTPC furniture; Epson MX-80 printer. Software. Call 219-261-2825.

•••

WANTED: Clock IC module for Motorola MEK6800D2: 614.4KHZ, IC part #MC6871B. Or will buy entire MEK6800D2 for parts. Frank Donnelly, 5 Peter Road, North Reading, Massachusetts 01864. Phone (617)664-4738 after 4PM EDT.

PT-68K-1 SINGLE BOARD COMPUTER

NEW!



NEW!

Peripheral Technology is announcing another innovative project. The PT-68K-1 utilizes the powerful 68008 processor for maximum performance at a price that is hard to beat. It features:

- 10 MHZ Clock
- Time of Day Clock
- 2 RS-232 Serial Ports
- 768K RAM
- Winchester Interface Capability
- Floppy Disk Controller
- No wait State RAM
- 2 8-Bit Parallel Ports
- 64K EPROM
- Board Size: 5.75" x 6"

OS9/68000 or SK Dos
Operating Systems Available

Price: $495.95

PERIPHERAL TECHNOLOGY
1480 Terrell Mill Rd., Suite 870
Marietta, Georgia 30067 USA
404/984-0742 Telex # 880584

OS9 is a trademark of Microware

LLOYD I/O™ INC.

Lloyd I/O is a computer engineering corporation providing software and hardware products and consulting services.

19535 NE GLISAN • PORTLAND, OR 97230 (USA)
PHONE: (503) 666-1097 • TELEX: 910 380 5448 LLOYD I O

Computer Engineers

K-BASIC™ *IS HERE*

K-BASIC is a TSC XBASIC (XPC) compatible COMPILER
for OS9 & FLEX... price $199

Here at last is a compiler for BASIC that will compile all your XBASIC programs. K-BASIC compiles TSC's XBASIC and XPC programs to machine code. K-BASIC is ready now to save you money and time by teaching your computer to:
• Think Faster • Conserve Memory • Be Friendlier

Call (503) 666-1097 for our CATALOG.
We have many programs for serious software developers!

DO™

Micro BASIC for OS9... $149

A structured mirco BASIC for general system control featuring: Parameter passing, 10 string variables, 26 numeric variables, subroutines, nested loops, interactive I/O, sequential files, and time variables (for applications executing in the background required to execute procedures such as disk or file backups) Includes the SEARCH and RESCUE UTILITIES™. (For OS9 ONLY.)

SEARCH and RESCUE UTILITIES™

for OS9... $35

A super directory search utility. Output may be piped to the included utilities to perform file: COPIES, DELETES, MOVES, LISTING (pagination), and FILTERING. Some filtering utility programs are included: of interest is the FILE DATE CHECKING utilities: YOUNGER and DRAFT (Level 2). (For OS9 Level 1 and 2.)

PATCH™

Modem Communications for OS9... $39

PATCH is a modem communications program for OS9 featuring: KEY MACROS, ASCII TEXT AND BINARY FILE UP/DOWN LOADING, PRINTER COPY, and HELP MENUS. We use it several times each day with our TELEX service. PATCH is convenient and easy to use. Key macros may be pre-stored and loaded at any time.

CRASMB™

CROSS ASSEMBLER PACKAGE
for OS9 & FLEX... all for $399
Motorola CPU's... $150
Intel CPU's... $150, Others... $150

CRASMB is the highly acclaimed cro[illegible]ge for OS9 and FLEX systems. It t[illegible] your [illegible] into a development statio[illegible] get C[illegible]s:

[illegible] 68[illegible] 6805 6809 6811 6502
[illegible] 8048 8051 80[illegible] Z80
(68000 [illegible]...$249)

CRASMB features [illegible] Conditi[illegible] Long symbol names, Symbol cross reference tab[illegible] code in 4 formats (OS9, FLEX, S-1-S9, INTEL HEX).

NOW FOR OS9/68K $432

VISA, MC, COD, CHECKS, ACCEPTED

USA:	LLOYD I/O (503 666 1097),	S.E. MEDIA (800 338 6800)
England:	Vivaway (0582 423425),	Windrush (0692 405189)
Germany:	Zacher Computer (65 25 299),	Kell Software (06203 6741)
Australia:	Ports Radio Electronics (344 9111)	
Japan:	Microboards (0474) 22-1741	Seikou (03) 832-6000
Switzerland:	Elsoft AG (056 86 27 24)	
Sweden:	Micromaster Scandinavian AG (018 - 138595)	

K-BASIC, DO, SEARCH and RESCUE UTILITIES
PATCH CRASMB and CRASMB 16.32 are trademarks of LLOYD I/O
OS9 is a ™ of Microware FLEX is a ™ of TSC

OS-9 UniFLEX
MUSTANG-020, 68020, 68881 AND MORE
HANDS-ON EXPERIENCE

The DATA-Comp Division of Computer Publishing Corporation announces their new and innovative HANDS-ON 68020 computer familiarization two day event. A chance to TRY BEFORE YOU BUY!

For two full days (Monday through Friday - excluding legal holidays) each participant will be furnished the exclusive use of a 68020 computer (MUSTANG-020). Each system will have available native C compilers, BASIC, assembler and other high level languages. Each system will be equipped with the Motorola MC 68881 math co-processor, where applicable.

Each demonstration room will contain not more than two work stations. Each system will be equipped with floppy disk, 20 megabyte winchester technology hard disk, and 2 megabyte of RAM. RAM is partitioned as 690K bytes of RAM disk and 1.2 megabyte of user RAM space.

Participants are encouraged to bring along any source level projects, for evaluation, in C, BASIC or assembler. Call for availibility of other HHLs.

Although this is not a training seminar, Data-Comp personnel are available for assistance and consultation. This event is scheduled for hands-on evaluations of the 68020 CPU, 68881 math co-processor and MUSTANG-020 system, operating in a functional environment.

Transportation to and from the airport and hotel/motel will be provided. Lunch provided both days. Chattanooga airport is serviced by American, Delta, Republic and other airlines.

COST

One person - $375.00
Two persons - $595.00

* Motel single $22.00, double $26.00
Includes satellite TV - convenient to food and shopping

DATA-COMP

A Division of
Computer Publishing, Inc.

5900 Cassandra Smith Road
Hixson, Tn 37343
Telephone 615 842-4600
Telex 510 600-6630

Systems available for both OS-9 and UniFLEX. Reservation should be made 15 days in advance. Attendee should initially indicate OS-9, UniFLEX or both. Special facilities available on request. Please write or call for additional information.

NOTE: Both OS-9 and UniFLEX are Unix type operating systems. Each as been enhanced in some aspect or another. Prospective attendees should have some working knowledge or experience with one of these operating systems, to gain full benefit of the session. However, a newcomer will find that it is a simple matter to be fairly proficient in using these systems in the allocated time. Special system instruction available on request. Call or write.

* Hotel/Motel cost are separate cost, not included in the basic cost shown.

THE 6800-6809 BOOKS

..HEAR YE.....HEAR

OS-9™ User Notes

By: Peter Dibble

The publishers of 68' Micro Journal are proud to make available the publication of Peter Dibbies OS9 USER NOTES

Information for the BEGINNER to the PRO, Regular or CoCo OS9

Using OS9
HELP, HINTS, PROBLEMS, REVIEWS, SUGGESTIONS, COMPLAINTS, OS9 STANDARDS, Generating a New Bootstrap, Building a new System Disk, OS9 Users Group, etc.

Program interfacing to OS9
DEVICE DESCRIPTORS, DIRECTORIES, "FORKS", PROTECTION, "SUSPEND STATE", "PIPES", "INPUT/OUTPUT SYSTEM", etc.

Programming Languages
Assembly Language Programs and Interfacing; Basic09, C, Pascal, and Cobol reviews, programs, and uses; etc.

Disks Include
No typing all the Source Listings in. Source Code and, where applicable, assembled or compiled Operating Programs. The Source and the Discussions in the Columns can be used "as is", or as a "Starting Point" for developing your OWN more powerful Programs. Programs sometimes use multiple Languages such as a short Assembly Language Routine for reading a Directory, which is then "piped" to a Basic09 Routine for output formatting, etc.

BOOK $9.95

Typeset -- w/ Source Listings
(3-Hole Punched; 8 x 11)
Deluxe Binder - - - - - - - - - - - $5.50

All Source Listings on Disk

1-8" SS, SD Disk - - - - $14.95
2-5" SS, DD Disks - - - $24.95

FLEX™ USER NOTES

By: Ronald Anderson

The publishers of 68 MICRO JOURNAL are proud to make available the publication of Ron Anderson's **FLEX USER NOTES**, in book form. This popular monthly column has been a regular feature in 68' MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. The most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

Listed below are a few of the TEXT files included in the book and on diskette.

All TEXT files in the book are on the disks.

LOGO.C1	File load program to offset memory — ASM PIC
MEMOVE.C1	Memory move program — ASM PIC
DUMP.C1	Printer dump program — uses LOGO — ASM PIC
SUBTEST.C1	Simulation of 6800 code to 6809, show differences — ASM
TERMEM.C2	Modem input to disk (or other port input to disk) — ASM
M.C2	Output a file to modem (or another port) — ASM
PRINT.C3	Parallel (enhanced) printer driver — ASM
MODEM.C2	TTL output to CRT and modem (or other port) — ASM
SCIPKG.C1	Scientific math routines — PASCAL
U.C4	Mini-monitor, disk resident, many useful functions — ASM
PRINT.C4	Parallel printer driver, without PFLAG — ASM
SET.C5	Set printer modes — ASM
SETBAS1.C5	Set printer modes — A-BASIC

NOTE: .CI,.C2, etc.=Chapter 1, Chapter 2, etc.

**Over 30 TEXT files included is ASM (assembler)-PASCAL-PIC (position independent code) TSC BASIC-C, etc.

Book only: **$7.95** + $2.50 S/H

With disk: 5" **$20.90** + $2.50 S/H

With disk: 8" **$22.90** + $2.50 S/H

Shipping & Handling $3.50 per Book, $2.50 per Disk set
Foreign Orders Add $4.50 Surface Mail
or $7.00 Air Mail
If paying by check - Please allow 4-6 weeks delivery
*** All Currency in U.S. Dollars**

Continually Updated In 68 Micro Journal Monthly

VISA

Computer Publishing Inc.
5900 Cassandra Smith Rd.
Hixson, TN 37343

(615) 842-4601
Telex 5106006630

*FLEX is a trademark of Technical Systems Consultants
**OS9 is a trademark of Microware and Motorola
***68' Micro Journal is a trademark of Computer Publishing Inc.

SK *DOS

(formerly called STAR-DOS)
is now available for both
68000 and 6809
computers. The same great DOS, but now better than ever, with enhancements which make it ideal for 6809 users moving to the 68000/68008/68010/68020. Available off-the-shelf now for the Emerald ESB-1 and Peripheral Technology PT-68K, and for licensing to OEMS at attractive terms. Single copies to end users are $75 (6809 version) and $125 (68K version). Configuration Manual (optional at $50) gives full details on adapting to new systems, supplied FREE to SK*DOS/68K purchasers until Dec. 1. Adapt SK*DOS to a new system and receive a royalty on your adaption! Call us at 914-241-0287 for more information.

STAR-KITS
SOFTWARE SYSTEMS CORPORATION
Box 209 Mt. Kisco NY 10549

ANDERSON COMPUTER CONSULTANTS

Ron Anderson, respected author and columnist for 68' Micro Journal announces the **Anderson Computer Consultants & Associates**, a consulting firm dealing primarily in 68XX(X) software design. Our wide experience in designing 6809 based control systems for machine tools is now available on a consultation basis.

Our experience includes programming machine control functions, signal analysis, multi-axis servo control (CNC) and general software design and development. We have extensive experience in instrumentation and analysis of specialized software. We support all popular languages pertaining to the 6809 and other 68XX(X) processors.

If you are a manufacturer of a control or measuring package that you believe could benefit from efficient software, write or call Ron Anderson. The fact is that any calculation you can do with a pencil and paper, can be done much better with a microcomputer. We will be happy to review your problem and offer a modern, state-of-the-art microcomputer solution. We can do the entire job orwork with your software or hardware engineers.

3540 Sturbridge Court
Ann Arbor, MI 48105

Hard Disk Subsystem for SS-50 Computers

THIS PROVEN SUBSYSTEM ADDS HARD DISK SPEED AND STORAGE CAPACITY TO YOUR COMPUTER YET REQUIRES ONLY ONE SS-30 SLOT. SOFTWARE (WITH SOURCE) IS INCLUDED FOR YOUR CHOICE OF FLEX9* OR SK*DOS*, OS-9* LEVEL I OR LEVEL II, OR OS-9 68K OPERATING SYSTEMS. THE SOFTWARE HONORS ALL OPERATING SYSTEM CONVENTIONS. THE SOFTWARE IS DESIGNED FOR THE XEBEX S1410 CONTROLLER INTERFACING TO ANY HARD DISK DRIVE THAT CONFORMS TO THE ST506 STANDARD. FOUR SUBSYSTEMS ARE AVAILABLE:

1) 27 MB (FORMATTED) CONTROL DATA CORPORATION WREN HARD DISK, XEBEC S1410A CONTROLLER, SS-30 INTERFACE CARD, ALL CABLES, AND SOFTWARE FOR $2850;
2) 7.3 MB (FORMATTED) TANDON TM-603 HARD DISK, REST SAME AS ABOVE FOR $895;
3) NO HARD DISK, REST SAME AS ABOVE FOR $600; AND
4) SS-30 INTERFACE CARD AND SOFTWARE FOR $200.

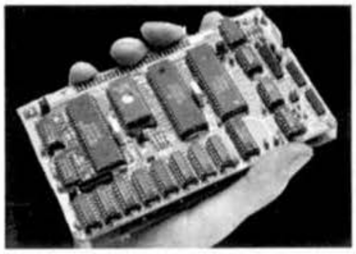
ALL PRICES INCLUDE SHIPPING. WE ACCEPT VISA AND MASTERCARD WITHOUT ADDING A SURCHARGE. TEXAS RESIDENTS MUST ADD SALES TAX. THE SUBSYSTEM MAY BE MOUNTEO WITHIN YOUR COMPUTER CHASSIS OR IN A SEPARATE ENCLOSURE WITH POWER SUPPLY. PLEASE WRITE OR PHONE (INCLUDE YOUR DAY AND EVENING PHONE NUMBERS) FOR MORE INFORMATION. WE WILL RETURN NORTH AMERICA CALLS SO THAI ANY DETAILED ANSWERS WILL BE AT OUR EXPENSE.

WELLWRITTEN™ ENTERPRISES

P.O. BOX 9802 - 845
AUSTIN, TEXAS 78766

VISA *** (512) 244-6530 *** MasterCard

FLEX IS A TRADEMARK OF TECHNICAL SYSTEMS CONSULTANTS, INC.
SK*DOS IS A TRADEMARK OF STAR-KITS
OS-9 IS A TRADEMARK OF MICROWARE AND MOTOROLA



512K RAM Expansion

Compact Flexible 6809 Computer

The new ST-2900 system — a complete 64K small business or hobbyist computer is only one of its many possible configurations. Among its features are:

- Small enough to hold in your hand! (Eurocard size: 3.9" x 6.3")
- Three board "system" for greater versatility than single board computers
- CPU Board — powerful 6809E processor, 16K or 64K RAM, 1K-32K EPROM, 2 RS232 serial ports with software programmable baud rates, 16 bit counter/timer. Run the CPU board all by itself, or plug your own custom board or our FDC board and/or RAM-512 board into the expansion connector.
- FDC Board — double-sided/double-density floppy disk controller with adjustment free digital data separator and write precompensation, 2 8-bit parallel ports, 2 16-bit counter/timers, prototyping area.
- **RAM-512 Board** — 524,288 bytes of RAM on a 4.15" x 6.3" board! Low power. Includes RAMDisk software for FLEX/STAR-DOS or OS-9.
- **FLEX, STAR-DOS,** and **OS-9** supported — **software selectable.**
- OS-9 Conversion Package lets you use the low cost Radio Shack CoCo version of OS-9 on our ST-2900 system. Save $131 off the suggested list price of OS-9! **No programming is involved.** Supports CoCo OS-9, standard OS-9, and MIZAR OS-9/68K disk formats. Compatible with PC-XFER to let you **read/write/format MS-DOS disks!**

CPU bare board plus EPROM	$45	OS-9 Conversion Package	$49
FDC bare board	$38	FLEX Conversion Package	$29
RAM-512 board A&T **(w/o RAM)**	**$299**	CPU + FDC + OS-9 Conversion	$119
CPU + FDC board set assembled and tested			$329

- Add $5 shipping/handling ($10 overseas). These prices are in U.S. funds. Canadian orders: call or write for prices. Terms: check, money order, VISA.

(trademarks: FLEX — Technical Systems Consultants. OS-9 — Microware & Motorola. MS-DOS — Microsoft)

Source code for ST-2900 OS-9 device drivers now available for $99

SARDIS TECHNOLOGIES

Call or write for free catalog and complete price list
(604) 255-4485

2261 E. 11th Ave. Vancouver, B.C., Canada V5N 1Z7

SOFTWARE FOR 680x AND MSDOS

SUPER SLEUTH DISASSEMBLERS
EACH $99-FLEX $101-OS/9 $100-UNIFLEX
OBJECT-ONLY versions: EACH $50-FLEX,OS/9,COCO
interactively generate source on disk with labels, include xref, binary editing
specify 6800.1,2,3,5,8,9/6502 version or Z80/8080,5 version
OS/9 version also processes FLEX format object file under OS/9
COCO DOS available in 6800.1,2,3,5,8,9/6502 version (not Z80/8080,5) only

CROSS-ASSEMBLERS (REAL ASSEMBLERS, NOT MACRO SETS)
EACH $50-FLEX,OS/9,UNIFLEX,MSDOS ANY 3 $100 ALL $200
specify for 180x,6502,6801,6804,6805,6809,Z8,Z80,8048,8051,8085,68000
modular cross-assemblers in C, with load/unload utilities and macros   NOW: OS-9-68K
8-bit (not 68000) sources for additional $50 each, $100 for 3, $300 for all

DEBUGGING SIMULATORS FOR POPULAR 8-BIT MICROPROCESSORS
EACH $75-FLEX $100-OS/9 $80-UNIFLEX
OBJECT-ONLY versions: E CH $50-COCO FLEX,COCO OS/9
interactively simulate processors, include disassembly formatting, binary editing
specify for 6800/1, (14)6805, 6502 6809 OS/9, Z80 FLEX

ASSEMBLER CODE TRANSLATORS FOR 6502, 6800/1, 6809

6502 to 6809	$75-FLEX $85-OS/9 $80-UNIFLEX
6800/1 to 6809 & 6809 to position-ind	$50-FLEX $75-OS/9 $60-UNIFLEX

FULL-SCREEN XBASIC PROGRAMS with cursor control
AVAILABLE FOR FLEX, UNIFLEX, AND MSDOS

DISPLAY GENERATOR/DOCUMENTOR	$50 w/source, $25 without
MAILING LIST SYSTEM	$100 w/source, $50 without
INVENTORY WITH MRP	$100 w/source, $50 without
TABULA RASA SPREADSHEET	$100 w/source, $50 without

DISK AND XBASIC UTILITY PROGRAM LIBRARY
$50-FLEX $30-UNIFLEX/MSDOS
edit disk sectors, sort directory, maintain master catalog, do disk sorts, resequence some or all of BASIC program, xref BASIC program, etc. non-FLEX versions include sort and resequence only

CMODEM TELECOMMUNICATIONS PROGRAM
$100-FLEX,OS/9,UNIFLEX,MS-DOS
OBJECT-ONLY versions: EACH $50
menu-driven with terminal mode, file transfer, MODEM7, XON-XOFF, etc. for COCO and non-COCO; drives internal COCO modem port up to 2400 Baud

DISKETTES & SERVICES

5.25" DISKETTES
EACH 10-PACK $12.50-SSSD/SSDD/DSDD
American-made, guaranteed 100% quality, with Tyvek jackets, hub rings, and labels

ADDITIONAL SERVICES FOR THE COMPUTING COMMUNITY

CUSTOMIZED PROGRAMMING
we will customize any of the programs described in this advertisement or in our brochure for specialized customer use or to cover new processors; the charge for such customization depends upon the marketability of the modifications.

CONTRACT PROGRAMMING
we will create new programs or modify existing programs on a contract basis, a service we have provided for over twenty years; the computers on which we have performed contract programming include most popular models of mainframes, including IBM, Burroughs, Univac, Honeywell, most popular models of minicomputers, including DEC, IBM, DG, HP, AT&T, and most popular brands of microcomputers, including 6800/1, 6809, Z80, 6502, 68000, using most appropriate languages and operating systems, on systems ranging in size from large telecommunications to single board controllers; the charge for contract programming is usually by the hour or by the task.

CONSULTING
we offer a wide range of business and technical consulting services, including seminars, advice, training, and design, on any topic related to computers; the charge for consulting is normally based upon time, travel, and expenses.

Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone 404-483-4570 or 1717

We take orders at any time, but plan
long discussions after 6, if possible.

Contact us about catalog, dealer, discounts, and services.
Most programs in source: give computer, OS, disk size.
25% off multiple purchases of same program on one order.
VISA and MASTER CARD accepted; US funds only, please.
Add GA sales tax (if in GA) and 5% shipping.
(UNIFLEX tm Technical Systems Consultants; OS/9 Microware; COCO Tandy; MSDOS Microsoft)

SOFTWARE FOR THE HARDCORE

** FORTH PROGRAMMING TOOLS from the 68XX&X **
** FORTH specialists — get the best!! **

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET COMPILATION for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware—

EPSON HX-20 rom system and target compiler
6809 rom systems for SS-50, EXORCISER, STD, ETC.
COLOR COMPUTER
6800/6809 FLEX or EXORCISER disk systems.
68000 rom based systems
68000 CP/M-68K disk systems, MODEL II/12/16

tFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTH™ is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

™ tFORTH and firmFORTH are trademarks of Talbot Microsystems
™ FLEX is a trademark of Technical Systems Consultants, Inc
™ CP/M-68K is trademark of Digital Research, Inc

tFORTH™
from TALBOT MICROSYSTEMS
NEW SYSTEMS FOR
6301/6801, 6809, and 68000

···> tFORTH SYSTEMS <···

For all FLEX systems: GIMIX, SWTP, SSB, or EXORcisor Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

** tFORTH — extended fig FORTH (1 disk) with fig line editor. $100 ($15)

** tFORTH + — more! (3 5" or 2 8" disks) $250 ($25)
adds screen editor, assembler, extended data types, utilities, games, and debugging aids.

** TRS-80 COLORFORTH — available from The Micro Works

** firm FORTH — 6809 only. $350 ($10)
For target compilations to rommable code.
Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include tFORTH +.

** FORTH PROGRAMMING AIDS — elaborate decompiler $150

** tFORTH for HX-20, in 16K roms for expansion unit or replace BASIC $170

** tFORTH/68K for CP/M-68K 8" disk system $290
Makes Model 16 a super software development system.

** Nautilus Systems Cross Compiler
— Requires: tFORTH + HOST + at least one TARGET:
— HOST system code (6809 or 68000) $200
— TARGET source code: 6800-$200, 6301/6801—$200
same plus HX-20 extensions— $300
6809—$300, 8080/Z80—$200, 68000—$350

Manuals available separately — price in ( ).
Add $6 system for shipping, $15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

AVAILABLE NOW!

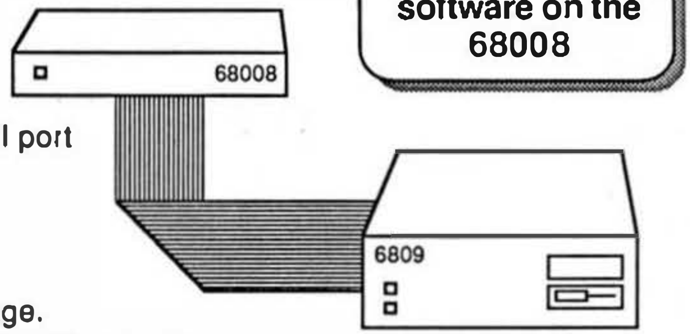
PLμS-68k (PL/9 for the 68000) running under FLEX™

- Built-in screen editor
- Built-in source-level debugger
- Byte, Integer and Long variables
- Single-pass compiler
- Direct source to object
- Compiles over 1000 lines/min
- Runs on any FLEX™ system with a spare PIA port

Develop 68000 software on your FLEX™ system. No second computer required!

The second processor module (included):

- 10MHz 68008 CPU
- 128k bytes RAM
- Case and power supply
- Plugs into Windrush UPROM-III port



Run FLEX™ software on the 68008

Other software included:

- Program loader
- 68000 FLEX™ interface package.
- Comprehensive System Monitor with source
- Hex-Binary-Hex Conversion Routines

Development is currently under way of a version for OS/9-68000. The package price includes a free copy of the OS/9 version when it becomes available.

$999
Complete

For further information, phone or write:

Worstead Laboratories
North Walsham
Norfolk NR28 9SA
England

Tel (44) 692 404086
Telex 975548 WMICRO G

WINDRUSH
Micro Systems Ltd.

6809<>68XXX UniFLEX

X-TALK

A C-MODEM/Hardware Hookup

Exclusive for the MUSTANG-020 running UniFLEX, is a new transfer program and cable set from DATA-COMP (CPI). X-TALK consist of 2 disks and a special cable, this hook-up enables a 6809 SWTPC UniFLEX computer to port UniFLEX files directly to a 68XXX UniFLEX system.

This is the only currently available method to transfer files, text or otherwise, from a 6809 UniFLEX system to a 68000 UniFLEX system, that we have seen. A must if you want to recompile or cross assemble your old (and valuable) source files to run on a 68000 UniFLEX system. GIMIX users can directly transfer files between a 6809 GIMIX system and our MUSTANG-020 68020 system, or GIMIX 68020 system. All SWTPC users must use some sort of method other than direct disk transfer. The 6809 SWTPC UniFLEX disk format is not readable by most other 68000 type systems.

The cable is specially prepared with internal connections to match the non-standard SWTPC SO/9 DB25 connectors. A special SWPTC+ cable and software is also available, at the same price. Orders must specify which type SWTPC 6809 UniFLEX system they intend to transfer from or to.

The X-TALK software is furnished on two disks. One 8" disk containing the 6809 software and one 5" disk containing the 68XXX software. These programs are also complete MODEM programs and can be used as such, including X-on X-off, and all the other features you would expect from a full modem program.

X-TALK can be purchased with/without the special cables, however, this SPECIAL price is available only to registered MUSTANG-020 owners.

X-TALK, w/cable	**$ 99.95**
X-TALK only	**69.95**
X-TALK w/source	**$149.95**

DATA-COMP
5900 Cassandra Smith Rd.
Hixson, TN 37343

Telephone 615 842-4601
Telex 510 600-6630

Note: Registered MUSTANG-020 owners must furnish system serial number in order to buy at these special low prices.

68' MICRO JOURNAL

Disk-1 Filesort, Minicat, Minicopy, Minifms, **Lifetime, **Poetry, **Foodlist, **Diet.
Disk-2 Diskedit w/ inst.& fixes, Prime, *Prmod, **Snoopy, **Football, **Hexpawn, **Lifetime
Disk-3 Cbug09, Sec1, Sec2, Find, Table2, Intext, Disk-exp, *Disksave.
Disk-4 Mailing Program, *Finddat, *Change, *Testdisk.
DISK-5 *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
Disk-6 **Purchase Order, Index (Disk file indx)
Disk-7 Linking Loader, Rload, Harkness
Disk-8 Crtest, Lanpher (May 82)
Disk-9 Datecopy, Diskfix9 (Aug 82)
Disk-10 Home Accounting (July 82)
Disk-11 Dissembler (June 84)
Disk-12 Modem68 (Revised June 86)
Disk-13 *Initmf68, Testmf68, *Cleanup, *Dskalign, Help, Date.Txt
Disk-14 *Init, *Test, *Terminal, *Find, *Diskedit, Init.Lib
Disk-15 Modem9 + Updates (Dec. 84 Gilchrist) to Modem9 (April 84 Commo)
Disk-16 Copy.Txt, Copy.Doc, Cat.Txt, Cat.Doc
Disk-17 Match Utility, RATBAS, A Basic Preprocessor
Disk-18 Parse.Mod, Size.Cmd (Sept. 85 Armstrong), CMDCODE, CMD.Txt (Sept. 85 Spray)
Disk-19 Clock, Date, Copy, Cat, PDEL.Asm & Doc., Errors.Sys, Do, Log.Asm & Doc.
Disk-20 UNIX Like Tools (July & Sept. 85 Taylor & Gilchrist). Dragon.C, Grep.C, LS.C, FDUMP.C
Disk-21 Utilities & Games - Date, Life, Madness, Touch, Goblin, Starshot, & 15 more.
Disk-22 Read CPM & Non-FLEX Disks. Fraser May 1986.
Disk-23 ISAM, Indexed Sequential file Accessing Methods, Condon Nov. 1985. Extensible Table Driven Language Recognition Utility, Anderson March 1986.
Disk-24 68' Micro Journal Index of Articles & Bit Bucket Items from 1979 - 1985, John Current.
Disk-25 KERMIT for FLEX derived from the UNIX ver. Burg Feb. 1986. (2)-5" Disks or (1)-8" Disk.
Disk-26 Compacta UniBoard Review, Code & Diagram. Burlinson March 1986.

NOTE:

This is a reader service ONLY! No warranty is offered or implied, they are as received by '68' Micro Journal, and are for reader convenience ONLY (some MAY include fixes or patches). Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

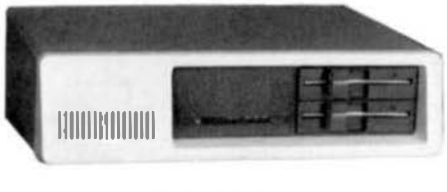
8" Disk $14.95 **5" Disk $12.95**

68' Micro Journal
5900 Cassandra Smith Rd. Hixson, TN 37343

☎
(615)-842-4600

Telex 5106006630

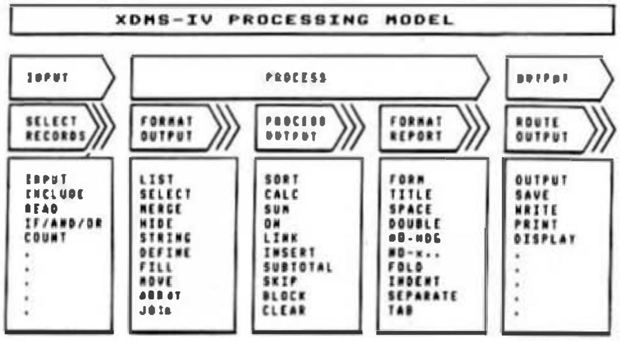
*Indicates 6800
**Indicates BASIC SWTPC or TSC
6809 no Indicator
Foreign Orders Add $4.50 for Surface Mail
or $7.00 for Air Mail
*All Currency in U.S. Dollars

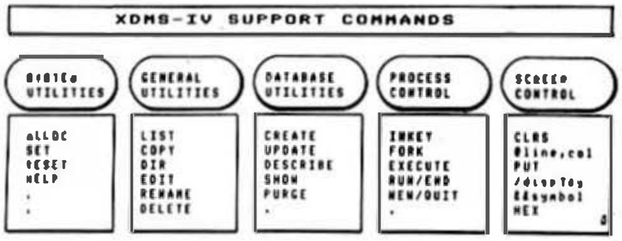


Winchester System

Custom Design Inquiries Welcome

Floppy System





GMX Micro-20 prices

MICRO 20 (12.5 MHz)	$2565.00
MICNO 20 (16.67 MHz)	$2895.00
8 PORT RS232 BOARD SET (SBC-8S)	$ 498.00
PROTOTYPING BOARD (SBC-WW)	$ 75.00
BACK PANEL PLATE (BPP-PC)	$ 44.00
I/O BUS ADAPTER (SBC-BA)	$ 195.00

QUANTITY DISCOUNTS ARE AVAILABLE ON THE ABOVE ITEMS AS FOLLOWS:4-9,LESS 5%; 10-24,LESS 10%; 25-99,LESS 20%; 100 UP,LESS 30%.

MC68881RC12	$ 295.00
MC68881RC16	$ 395.00
SBC ACCESSORY PACKAGE (M20-AP)	$1690.00

For other configurations and options, contact GMX.

MOTOROLA 68020 USERS MANUAL	$ 10.00
MOTOROLA 68881 USERS MANUAL	$ 10.00

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00. Foreign orders over $200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account number 73-32033.

BASIC-09 and OS-9 are trademarks of Microware Systems Corp. and MOTOROLA, Inc. FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc. GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.

GMX

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609

(312) 927-5510 • TWX 910-221-4055

GMX S-50 BUS prices

68020 SYSTEM

for the user who appreciates the need for a bus structured system using STATIC RAM and powered by a ferro resonant constant voltage transformer, DMA transfers, high speed MMU, we have the UNIFLEX-VM 68020 development system.
The system CPU provides protection to the system and other users from crashes caused by defective user programs.
The system's Intelligent serial I/O processor boards significantly reduce system overhead by handling routine I/O functions.
The UniFLEX VM Operating System Is a demand-paged, virtual memory operating system written In 68020 Assembler code for compactness and efficiency. It allows up to 4 Megabytes of Virtual Memory per user. All systems Include 1MB of static RAM, one 3-port Intelligent Serial I/O board, DMA Controllers, a 5" 80 track floppy drive.

PRICES

#020 UniFLEX VM with 25MB HD	$10,980.20
#020 UniFLEX VM with 85MB HD	$12,480.20

YOU CAN EXPAND THESE 020 SYSTEMS WITH;

60MB STREAMER	$ 2,400.00
REMOVABLE PACK DRIVE	$ 1,200.00

INTELLIGENT I/OS

#14 3 Port Serial-30 Pin	$ 498.14
#13 4 Port Serial-50 Pin	$ 618.13
#12 Parallel-50 Pin	$ 538.12

CABLE SETS FOR I/OS

# 95 Cable Sets Specify Card	$ 24.95
# 51 Cent. B.P. Cable for #12 & #44	$ 34.51
# 53 Cent. Cable Set	$ 36.53

6809 SYSTEM

The number 39 systems Include:#05 CPUwDAT; #19 Classy Chassis; 256K Static RAM; a # 43 2 port serial card & cables; #68 DMA Controller; all necessary cables, power regulators, and filler plates;

System # 39 OS-9 GMX II Dual 80 DSDD		$ 2,998.39
•	w19MB	$ 4,698.39
	w72MB	$ 6,298.39

The Software included in this System: GMXBUG monitor; FLEX; and OS-9 GMXII. You can software select either FLEX or OS-9. Also includes OS-9 Editor, Assembler, Debugger, BASIC-09, RUMB, RMS, DO, and GMX-VDISK for FLEX,

System # 39 UniFLEX	w25MB	$ 4,698.39
•	w85MB	$ 6,298.39

The UniFLEX Operating System is Included.

6809 SYSTEMS USING THE GIMIX III CPU & INTELLIGENT I/O PROCESSOR BOARDS

These System Include: GMX6809 CPU III; one #11 3 port Intelligent serial I/O & Cables; #19 Classy Chassis; 256K Static RAM; #68 DMA controller; all necessary cables, power regulators, and filler plates.

System # 79 OS9 GMX III Dual 80 DSDD		$ 4,498.79
•	w25MB	$ 6,498.79
•	w85MB	$ 7,998.79

The # 79 System Software includes: OS9 GMXIII; OS9 Editor, Assembler, Debugger, BASIC 09, RUXB, RMS, DO, RAMdisk, O-FLEX; GMXBUG; FLEX. The GMX Support ROM and the hardware CRC board are exclusive features included in this system.

System # 89 UniFLEX III	w25MB	$ 6,798.39
	w85MB	$ 8,298.39

The UniFLEX GMX III Operating System is Included.

For Ordering Call

(615)842-4600

FROM – DATA-COMP, C.P.I.

Now Offering: *FLEX™ (2 Versions)
AND *STAR-DOS PLUS+™

A Family of 100% 68XX Support Facilities
The Folks who FIRST Put FLEX™ on
The CoCo

FLEX-CoCo Sr.
with TSC Editor
TSC Assembler
Complete with Manuals
Reg. $250.⁰⁰ **Only $79.⁹⁵**

STAR-DOS PLUS+
- Functions Same as FLEX
- Reads - writes FLEX Disks $34.⁹⁵
- Run FLEX Programs
- Just type: Run "STAR-DOS"
- Over 300 utilities & programs to choose from.

FLEX-CoCo Jr.
without TSC
Editor & Assembler
$49.⁹⁵

PLUS

ALL VERSIONS OF FLEX & STAR-DOS* INCLUDE

TSC Editor
Reg $50.00
NOW $35.00

+ Read-Write-Dir RS Disk
+ Run RS Basic from Both
+ More Free Utilities
+ External Terminal Program
+ Test Disk Program
+ Disk Examine & Repair Program
+ Memory Examine Program
+ Many Many More!!!

TSC Assembler
Reg $50.00
NOW $35.00

CoCo Disk Drive Systems

2 THINLINE DOUBLE SIDED DOUBLE DENSITY DISK DRIVES SYSTEM WITN POWER SUPPLY,CABINET,DISK DRIVE CABLE,J&M NEW DISK CONTROLLER JPD-CP WITH J-DOS,RS-DOS OPERATING SYSTEMS.	$469.95

* Specify What CONTROLLER You Went J&M, or RADIO SHACK

THINLINE DOUBLE SIDED DOUBLE DENSITY 40 TRACKS	$129.95

Verbatim Diskettes

Single Sided Double Density	$ 24.00
Double Sided Double Density	$ 24.00

Controllers

J&M JPD-CP WITH J-DOS	$139.95
WITH J-DOS, RS-DOS	$159.95
RADIO SHACK 1.1	$134.95
RADIO SHACK Disk CONTROLLER 1.1	$134.95

Disk Drive Cables

Cable for One Drive	$ 19.95
Cable for Two Drives	$ 24.95

MISC

64K UPGRADE FOR C,D,E,F, AND COCO II	$ 29.95
RADIO SHACK BASIC 1.2	$ 24.95
RADIO SHACK DISK BASIC 1.1	$ 24.95
DISK DRIVE CABINET FOR A SINGLE DRIVE	$ 49.95
DISK DRIVE CABINET FOR TWO THINLINE DRIVES	$ 69.95

PRINTERS

EPSON LX-80	$289.95
EPSON MX-70	$125.95
EPSON MX-100	$495.95

ACCESSORIES FOR EPSON

8148 2K SERIAL BOARD	$ 89.95
8149 32K EXPAND TO 128K	$169.95
EPSON MX-RX-80 RIBBONS	$ 7.95
EPSON LX-80 RIBBONS	$ 5.95
TRACTOR UNITS FOR LX-80	$ 39.95

CABLES & OTHER INTERFACES
CALL FOR PRICING

DATA-COMP
5900 Cassandra Smith Rd.
Hixson, TN 37343

SHIPPING
USA ADD 2%
FOREIGN ADD 5%
MIN. $2.50

☎
(615)842-4600
For Ordering
Telex 5106006630

Introducing

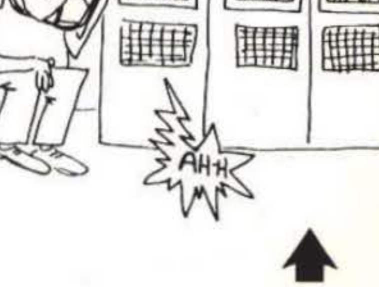
S - 50 BUS / 68XX

Board and/or Computer
Terminals-CRTs-Printers
Disk Drives-etc.

REPAIRS

This

Not This



NOW AVAILABLE TO ALL S50/68XX USERS

The Data-Comp Division of CPI is proud to announce the availability of their service department facilities to **'ALL'** S50 Bus and 68XX users. Including all brands, SWTPC - GIMIX - SSB - HELIX and others, including the single board computers. *Please note that kit-built components are a special case, and will be handled on an individual basis, if accepted.

1. If you require service, the first thing you need to do is call the number below and describe your problem and *confirm a Data-Comp service & shipping number!* This is very important, Data-Comp will not accept or repair items not displaying this number! Also we cannot advise or help you troubleshoot on the telephone, we can give you a shipping number, but **NO** advice! Sorry!

2. All service shipments must include both a minimum $40.00 estimate/repair charge and pre-paid return shipping charges (should be same amount you pay to ship to Data-Comp).

3. If you desire a telephone estimate after your repair item is received, include an additional $5.00 to cover long distance charges. Otherwise an estimate will be mailed to you, if you requested an estimate. Estimates *must be requested.* Mailed estimates slow down the process considerably. However, if repairs are not desired, after the estimate is given, the $40.00 shall constitute the estimate charge, and the item(s) will be returned unrepaired providing sufficient return shipping charges were included with the item to be serviced. Please note that estimates are given in dollar amounts only.

4. Data-Comp service is the oldest and most experienced general S50/68XX service department in the world. We have over $100,000.00 in parts in stock. We have the most complete set of service documents for the various S50/68XX systems of anyone - **YET, WE DO NOT HAVE EVERYTHING!** But we sure have more than anyone else. We repair about 90% of all items we receive. Call for additional information or shipping instructions.

DATA-COMP
5900 Cassandra Smith Rd.
Hixson, TN 37343

☎
(615)842-4607
Telex 5106006630